Depression Detection Via Twitter

Thesis submitted for the degree of
Master of Philosophy
at the University of Leicester

by

Lei Tong
Biomedical Image Processing Lab
School of Informatics
University of Leicester

March 25, 2020
Declaration of Authorship

I, Lei TONG, declare that this thesis titled, “Depression Detection Via Twitter” and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.

- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.

- Where I have consulted the published work of others, this is always clearly attributed.

- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.

- I have acknowledged all main sources of help.

- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed:
__________________________________________

Date:
__________________________________________
Depression is one of the most common mental health disorders, and a large number of depressed people commit suicide each year. Potential depression sufferers do not consult psychological doctors because they feel ashamed or are unaware of any depression, which may result in severe delay of diagnosis and treatment. In the meantime, evidence shows that social media data provides valuable clues about physical and mental health conditions. In this thesis, I argue that it is feasible to identify depression at an early stage by mining online social behaviours. My approach, which is innovative to the practice of depression detection, does not rely on the extraction of numerous or complicated features to achieve accurate depression detection. Instead, I propose a novel classifier, namely, Inverse Boosting Pruning Trees (IBPT), which demonstrates a strong classification ability on two publicly accessible Twitter depression detection datasets. Subsequently, I propose a novel deep neural topic model, FastText Distributional Prior DocNADE with Attention Mechanism (fdp-DocNADEa) that collects interpretable topics from Twitter users’ posting content and generates representative document features for depression classification. To comprehensively evaluate the clustering capability of the fdp-DocNADEa, I use three real text datasets and the fdp-DocNADEa still obtains competitive results against several state of the arts techniques. Finally, I combine the two newly proposed methods to form an entire framework for Twitter depressed users classification and the results manifest that my proposed framework is promising for identifying Twitter users with depression.
Acknowledgements

I would like to express my sincere gratitude to the following people for their generous help and support over the two years.

First of all, I would like to express my gratitude and appreciation to my first supervisor, Dr Huiyu Zhou for his patience, valuable guidance and constructive research suggestions from the beginning to completion of this MPhil study. Working with Dr Zhou is a wonderful experience. Dr Zhou’s teaching style can be summarized by a Chinese proverb ‘Give a man a fish and you feed him for a day. Teach him how to fish and you feed him for a lifetime’. When I met bottle-necks in my project, he did not give me the answers to solve the problems directly. Instead, he advised me to decompose the problems into a number of sub-problems and then address them one by one. This practice was beneficial as it helped improving my capacity of independent work. His rigorous academic attitude has inspired me during my MPhil, and will have a profound influence on my academic career forever. Moreover, he guided me how to write scientific papers and look for research problems. The MPhil research could not be completed without his guidance, patience and encouragement.

I would like to thank my second supervisor, Dr Qian Gong. I would say she provided emotional supports. Her professional research experience on social networks also inspires my works.

Besides, I would like to thank my lab colleagues, Zhihua Liu, Zheheng Jiang and Long Chen. They helped me to address the technical problems of my research project, such as code debugging and experimenting. Through communication and exchange with them, I have learnt how to organise my research work and collaborate with others to solve research and development problems.

Finally, I would like to thank my parents for their moral and financial support, especially unconditional love at all times. They always have a special space in my heart.
## Contents

Declaration of Authorship ........................................... ii
Abstract ........................................................................ iii
Acknowledgements ..................................................... iv

1 Introduction .................................................................. 1
   1.1 Research Background ............................................. 1
   1.2 Aim and Objective ................................................ 3
   1.3 Contributions ........................................................ 4
   1.4 Thesis Outline ....................................................... 5
   1.5 Publication List ..................................................... 6

2 Literature Review ...................................................... 7
   2.1 Value of Online depression detection ....................... 7
   2.2 Online depression detection based on Statistical methods 8
   2.3 Online depression detection based on Natural Language Processing methods ......................................................... 9
   2.4 State of the art online depression detection frameworks 12

3 Inverse boosting pruning trees for depression detection on Twitter ......................................................... 15
   3.1 Introduction .......................................................... 15
   3.2 Review of recent boosting methods ............................ 16
   3.3 Proposed Method .................................................... 18
       3.3.1 Inverse boosting pruning trees ............................ 18
           Search for the best pruned tree ............................... 19
           Inverse boosting structure .................................... 21
       3.3.2 Convergence Analysis ....................................... 22
   3.4 Experiment Setup .................................................. 25
       3.4.1 Datasets ........................................................ 25
List of Figures

3.1 Proposed framework. ...................................................... 16
3.2 The error rate curve of the training and validation sets corresponding to the pruning process. ...................................................... 21
3.3 This figure shows the IBPT’s error rate, estimator error, estimator weight changing with the number of the estimators against the depression dataset: (a) Test error curve in training. (b) Estimator error curve. (c) Estimator weight curve. ...................................................... 30
3.4 Feature importance list by IBPT for three toy datasets: (a) LSVT. (b) News. (c) Glasses. ...................................................... 35
3.5 Joint distributions against features using kernel density estimation (KDE), where contours represent the distribution of features and samples density become high with the contours color burning: (a) LSVT - more important features. (b) LSVT - less important features. (c) News - more important features. (d) News - less important features. (e) Glasses - more important features. (f) Glasses - less important features. ...................................................... 36
3.6 Important feature list in the Twitter dataset. (a) The TTDD dataset. (b) The CLPsych dataset ...................................................... 38
3.7 Salient words by LDA: (a) Normal class topic words. (b) Depression class topic words. ...................................................... 39
3.8 Twitter set - features joint distributions using kernel density estimation (KDE), where contours represent the distribution of features and samples density become high with the contours color burning: (a) Sample distribution against retweet counts and negative words. (b) Sample distribution against total favourites and emoticons. ...................................................... 39
3.9 Statistics of online behaviours of depression and non-depressed users. (a) Active time comparison. (b) Posting custom of each tweet. (c) Statistics of user profile features. ............................. 40

3.10 The CLPsych dataset - features joint distributions using kernel density estimation (KDE), where contours represent the distribution of features and samples density become high with the contours color burning: (a) Sample distribution against retweet counts and negative words. (b) Sample distribution against total favourites and emoticons. ................................................. 41

3.11 Statistics of online behaviours of depression, ptsd and control users on the CLPsych dataset. (a) Active time comparison. (b) Posting custom of each tweet. (c) Statistics of user profile features. ....... 42

3.12 ROC curves of different feature combinations. (a) The TTDD dataset.(b) The CLPsych dataset. ................................................................. 42


4.2 Replicated Softmax Model (RSM). Image Courtesy of [32] ............ 48

4.3 Document Neural Autoregressive Topic Model (DocNADE). Blue line represents hidden vectors $h_t$ share same weight in the generation of output. Image Courtesy of [44] ................................. 49

4.4 Schematic overview of fdp-DocNADE. The input and output (double circle) of each word $v_d$ are multinomial. Left green matrix $W$ represents the initial word embeddings and right red matrix $E$ is the introduced FastText pre-trained word embeddings. $w_d$ and $e_d$ are embedding vectors looking up from $W$ and $E$ respectively. $h_d^{\rightarrow}$ denotes the forward network projected hidden units (latent topics) from the concat of $w_d$ and $e_d$. Blue dot lines with dots signify the shared weight of hidden units. .................................................... 51

4.5 Schematic overview of fdp-DocNADEa. I add an attention layer between hidden layer and output to reassign the weight of each unit $h_d^{\rightarrow}$ based on syntactic information. Grey line $a_d$ is the reassigned weight for each hidden units $h_d^{\rightarrow}$. $C_d$ denotes a context vector which includes context information of $h_d$ hidden unit ......... 54

4.6 One word “attends” to other words in the same sentence differently. Image courtesy of [82] .................. 54
4.7 Evaluating the F1 score and the accuracy with the different training data percentages of the TTDD dataset (a) The F1 score curve (b) The accuracy curve. ............................... 62

4.8 Evaluating the F1 score and the accuracy with the different training data percentages of the CLPsych dataset (a) The F1 score curve (b) The accuracy curve. ............................... 63

4.9 Document representations distribution using TSNE on TTDD dataset. (a): Document vectors learned from LDA. (b): Document vectors learned from DocNADE. (c): Document vectors learned from fdp-DocNADE. (d): Document vectors learned from fdp-DocNADEa. 64


4.11 (a) TTDD set: Top four topics with highest average difference absolute weight between non-depressed and depressed class. (b) CLPsych set: Top four topics with highest average difference absolute weight between control, depression and ptsd users .................. 66

4.12 Top probability words in four topics on the TTDD set with manual defined ‘theme’ .................................................. 67

4.13 Top probability words in four topics on the CLPsych set with manual defined ‘theme’ ............................................. 68

4.14 Roc curve of the Inverse Boosting Pruning Trees training with topic models document representations (a) Roc of the TTDD set (b) Roc of the CLPsych set .............................................. 69
List of Tables

2.1 LIWC Evaluation Results. I-WORDS represents the percentage of first person words in whole sentence. Similarly, SOCIAL WORDS count the number of social words. POSITIVE EMOTIONS and NEGATIVE EMOTIONS compute the polarity score of input sentence, ranging from 0 to 100. COGNITIVE PROCESSES also return a value to represent sentence cognition. ............................. 10

2.2 Summary of methods for Social networks users depression classification. ................................................................. 14

3.1 Twitter depression dataset Statistics. .................................................. 27

3.2 TTDD Dataset. The evaluation criterion is Area Under the curve of ROC (AUC) score. .................................................. 32

3.3 The CLPsych Dataset: The evaluation criterion is Area Under the curve of ROC (AUC) score. ........................................... 32

3.4 LSVT Dataset. The evaluation criterion is Area Under the curve of ROC (AUC) score. .................................................. 33

3.5 News Dataset. The evaluation criterion is Area Under the curve of ROC (AUC) score. .................................................. 34

3.6 Glass Dataset. The evaluation criterion is Area Under the curve of ROC (AUC) score. .................................................. 34

3.7 Feature Statistics of the CLPsych dataset. The row of LDA topics present the top-10 most salient words in each class. The remain rows collect the average value of features in three classes. ........ 41

3.8 Classification comparison with SOTA methods using F1 score and accuracy measures. ................................................. 43
4.1 Data Statistics. The column of Train Validation and Test represent
the training, testing and validation set size of each database. The
symbol V: Vocabulary Size, L: Average Text length, C: Number of
Classes. Domain means the data source of each database. . . . . . . . 57
4.2 Statistics of perplexity value. LDA and HDP belong to traditional
machine learning topic model. The remaining methods are in-
cluded in deep learning based topic models. . . . . . . . . . . . . . 58
4.3 Top coherence score topics with top 10 words in 20NS . . . . . . . 60
4.4 Statistics of average topic coherence with top ten (W10) and twenty
(W20) words. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 60
4.5 Text classification with F1 score and accuracy measures. . . . . . 61
4.6 Classification comparison with SOTA methods using F1 scores and
accuracy measures. . . . . . . . . . . . . . . . . . . . . . . . . . . . . 70
List of Abbreviations

AI  Artificial Intelligence
NLP  Natural Language Processing
MDD  Major Depression Disorder
SVM  Support Vector Machine
Adaboost  Adaptive Boosting
IBPT  Inverse Boosting Pruning Trees
TTDD  Tsinghua Twitter Depression Database
CLPsych  CLPsych 2015 Twitter Dataset
KDE  kernel density estimation
AUC  Area under the curve (Receiver operating characteristic)
LDA  Latent Dirichlet Allocation
BoW  Bag of Words
DocNADE  Neural Autoregressive Topic Model
fdp-DocNADE  FastText Distributional Prior DocNADE
fdp-DocNADEa  FastText Distributional Prior DocNADE with the Attention Mechanism
MCMC  Markov chain Monte Carlo
MLP  Multilayer Perceptron
CSE-D  Center for Epidemiological Studies Depression Scale
LIWC  Linguistic inquiry and word count
STM  Social Network Mental Detection-base Tensor Model
API  Application programming interface
LR  Logistic Regression
PCA  Principal Component Analysis
HDP  Hierarchical Dirichlet Process
LSA  Latent Semantic Analysis
SOTA  State of the art
GBDT  Gradient Boosting decision trees
<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RF</td>
<td>Random Forest</td>
</tr>
<tr>
<td>Ada_NB</td>
<td>Adaboost with Naive Bayes</td>
</tr>
<tr>
<td>Ada_SVM</td>
<td>Adaboost with support vector machine</td>
</tr>
<tr>
<td>Ada_PT</td>
<td>Adaboost with pruning trees</td>
</tr>
<tr>
<td>KDE</td>
<td>kernel density estimation</td>
</tr>
<tr>
<td>ROC</td>
<td>Receiver operation characteristic</td>
</tr>
<tr>
<td>F1</td>
<td>F1 score</td>
</tr>
<tr>
<td>acc</td>
<td>Accuracy</td>
</tr>
<tr>
<td>RSM</td>
<td>Replicated Softmax Model</td>
</tr>
<tr>
<td>NVDM</td>
<td>Neural Variational Document model</td>
</tr>
<tr>
<td>TMN</td>
<td>TagMyNews Dataset</td>
</tr>
<tr>
<td>20NS</td>
<td>20 Newsgroup dataset</td>
</tr>
<tr>
<td>TSNE</td>
<td>t-Distributed Stochastic Neighbor Embedding</td>
</tr>
<tr>
<td>PTSD</td>
<td>Post-traumatic stress disorder</td>
</tr>
<tr>
<td>iDocNADEe</td>
<td>Document Informed Neural Autoregressive Distribution Estimator with Embeddings</td>
</tr>
<tr>
<td>ctx-DeepDNEe</td>
<td>Deep Contextualized DocNADE with Distributional Semantics</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

1.1 Research Background

Depression is one of the most common mental illnesses. It is estimated that nearly 360 million people suffer from depression [52]. British Mental Health Foundation [52] made statistics of the most common mental health problems in 2016: (1) In Britain, 7.8% of people meet the criteria of depression diagnosis. (2) 4 ~ 8 % of British citizens will experience depression in their lifetime. (3) Mixed anxiety and depression has been estimated to cause one fifth of people to lost work in Britain. Andrade et al. [3] reported that the probability for an individual to encounter a major episode of depression within a period of one year was 3-5% for males and 8-10% for females. Although neural therapy or psychological guidance can ease the depression disorder, this disorder will relapse and remission in the lifetime of patients. Because of depression, about one million of people committed suicide annually in the world [15].

National Collaborating Centre for Mental Health (Great Britain) reported [53] that the vast majority (up to 90%) of depressive and anxiety disorders diagnosed are treated in primary care. Depressed people may have a variety of symptoms: having troubles in going to sleep or sleeping too much, lacking of passion, feeling hopeless or disappointed [67]. In clinical exercises, psychological specialists are looking for reliable methods to detect and prevent depression. Yang et al. [83] investigated the relation between vocal prosody and changes in depression severity over time. Alghowinem et al. [2] examined human behaviours such as speaking behaviours and eye activities associated with major depression. Kulkarni et al. [41] made statistics of Major Depression Disorder sufferers’ symptoms and summarized that an individual may be suffering from depression if they show signs
of lacking of passion in favourite activities, extreme anger, constant regret, thinking of suicide without a clear physical cause over a period of at least two weeks. Diagnostic and Statistical Manual of Mental Disorders [73] is an important reference for psychological doctors to diagnose depression. There are nine classes of depression symptoms reported in the document, describing the distinguishing behaviours in patients’ daily life. Nevertheless, the symptoms of depression disorders evolve over time and it has been advised to dynamically update the criteria of depression clinically [15].

Depression sufferers who do not receive timely treatment will develop worse conditions. More than 70% of people in the early stage of depression did not consult psychological doctors, and their conditions deteriorated [73]. Depression if left untreated may lead to devastating consequences apart from the disease itself, such as increased chances of risky behavior like drug or alcohol addiction, making it difficult to recover from other serious illnesses and the disease itself may be prolonged for years [41]. González-Ibáñez et al. [28] reported that people are somehow ashamed or unaware of depression which makes them miss timely treatment. Britain NHS [53] reported general practitioners’ (GP) cognition is poor on recognizing depression disorders, and only a few individuals who experience depression disorder ever get treatment. The problem is caused by GPs’ difficulties in recognising the disorder, but it is also stemming from patients’ worries about stigma. Thus, an advanced depression diagnosis method that cannot only identify depression sufferers at an early stage but also protect individuals’ privacy is urgently needed.

Choudhury et al. [21] and Neuman et al. [58] proposed to explore the correlation of depression sufferers with their online behaviours on social networks. With the explosive growth of computer network applications, social networks have become an indispensable part of many people’s daily lives. 62% of the American adults (age 18 and older) use Facebook, whilst the majority of the users (70%) visit Internet daily and a large portion of the users access to Internet multiple times each day [85]. There are 1.10 billion posts on Facebook every day. Twitter and Tumblr also have 500 and 77.5 million users who are active per day, where 70% of the Twitter users log in every day [85]. In many cases people would prefer to share their emotional experience and daily encounter online rather than communicate with friends offline. As such, social networks provide means for capturing behavioural attributes that are relevant to an individual’s thinking, mood,
1.2. Aim and Objective

In this research project, I intend to develop a new automated system to determine whether or not a social media user suffers from depression through monitoring his/her Twitter data. To do so, I will extract corresponding features from Twitter data logs, such as posting time, polarity of text content. Then, machine learning techniques will be developed to detect and report the mental condition of the user.

The existing online depression frameworks \([61, 20, 74, 73]\) consist of two primary procedures, extracting representative features of depressed users’ online behaviours and classifying them by using machine learning classifiers. In feature extraction, the state of the art \([21, 74, 73]\) methods mostly focus on exploring new features from users’ posting content by employing natural language processing (NLP) methods such as sentiment analysis, topic modelling. But the employed NLP approaches are not the latest techniques which have some aspects can be improved. Besides, the established classifiers from previous researches \([73, 57, 21]\) for online depression detection cannot achieve consistent performance due to the influence of noisy data. They have not been tested over their robustness in different environments. I aim to establish my novel depression detection framework to address the above challenges. I discuss the details of existing online depression detection methods in Chapter 2 and the objectives of my study are summarized as follows:
Chapter 1. Introduction

(1) The main objective of this project is to develop a novel framework using machine learning techniques to establish the relationship between social media messages and depression.

(2) The previously developed classifiers cannot achieve consistent performance on complex social media datasets, so my second objective is to develop a novel classification algorithm which has a strong generalisation ability to extract meaningful social media features and separate depressed users and normal users.

(3) My third objective is to develop a novel NLP method to extract representative document topic features by incorporating deep learning and help readers to understand the posting content of depressed people.

(4) The final objective is to evaluate the proposed methods over different datasets.

1.3 Contributions

In this thesis, my research progress consist of two stages. In the first stage of research, my main contribution is establishing an automated Twitter depression detection system which is presented in Chapter 3. The details of my new contribution in the first stage is reported as follows:

(1) I develop a novel classifier, namely Inverse Boosting Pruning Trees (IBPT) based boosting algorithm, which combines multiple weak models with the power of Adaboost in pruning decision trees. The IBPT outperforms several baselines on a publicly accessible depression database. I also conduct experiments to evaluate the classification capability of the IBPT on three real datasets from UCI machine learning repository and the IBPT still achieves the best classification result.

(2) I study user profile, social interaction and linguistic features extracted from the Twitter dataset and evaluate the importance of these features in the framework of the IBPT. The weighting of these features is optimally adapted so that the features can effectively contribute to the classification outcome.

(3) I conduct comprehensive experiments to justify the significance of the proposed strategy over heterogeneous databases.
In the second stage of the project, I make contributions to collect interpretable topics from Twitter users’ posting content and generate representative document features for depression users’ classification. In this stage, my new contribution is summarized below:

(1) I propose a novel deep neural topic model, namely FastText Distributional Prior DocNADE (fdp-DocNADE) based on Document Neural Autoregressive Topic Model (DocNADE), which combines DocNADE with the word semantic embeddings of the FastText distribution prior. I conduct experiments to evaluate the interpretability and applicability of the proposed model on five databases including three publicly accessible databases and two Twitter depression detection databases, and fdp-DocNADE shows the ability of producing coherent topics and can produce representative document vectors for document classification.

(2) I present an improved method based on fdp-DocNADE, named FastText Distributional Prior DocNADE with the Attention Mechanism (fdp-DocNADEa) incorporating the Attention Mechanism with fdp-DocNADE to extract syntactic features from the documents. The proposed model performs well on the evaluation of model generalisation ability.

(3) I conduct comprehensive experiments to justify the significance of my two proposed models on five datasets, which show competitive results in comparison with several state of the art methods.

(4) In my Twitter datasets, I employ fdp-DocNADEa to discover typical topics from the depressed and non-depressed user groups. I verified that my online depression detection framework achieved promising performance.

1.4 Thesis Outline

In this section, I provide an overview of the thesis chapters. In Chapter 1, I give an introduction of my research background and present my research objectives and new contributions.
Chapter 1. Introduction

In Chapter 2, I review the literature about the recent depression analysis studies and state the strength and weakness of the existing methods for online depression detection. I also introduce the details of two public social media depression detection datasets.

In Chapter 3, I introduce my first contribution for online depression detection. In this Chapter, I give detailed introduction to my established depression classification system and verify the effectiveness of my proposed methods. Besides, I make statistics of Twitter features to discover the difference of online behaviours between normal and depressed users.

In Chapter 4, I propose a novel topic model to extract interpretable and representative topic features from documents and I evaluate the effectiveness of my proposed model for depression classification.

In Chapter 5, I conclude my works in this thesis and discuss my research limitations then present potential perspective of further research events.

1.5 Publication List

The content of Chapter 3 appear in:


Non-thesis research: I have also contributed to the following publications:


Chapter 2

Literature Review

2.1 Value of Online depression detection

Most people feel sad or depressed at times. It is a normal reaction to loss or life’s struggles. But when intense sadness including feeling helpless, hopeless, and worthless lasts for many days to weeks and keeps people from living their life, it may be something more than sadness. It could be clinical depression, a treatable medical condition [67]. Early detecting whether or not people have trend of suffering depression is benefit for treatment [62]. DSM-5[4] is an important reference for doctors to diagnose depression disorders and some typical depression symptoms are reported in the document:

- A depressed mood during most of the day, especially in the morning.
- Feel tired or lack of energy almost every day.
- Can not sleep or sleep too much almost every day.
- Think often about death or suicide (not just a fear of death).

While these symptoms are common, not everyone with depression will has the same ones. How severe they are, how often they happen, and how long they last can vary. The diagnosis of depression is not generalized and almost depends on the doctor’s clinical experience [28, 73]. Psychological specialists are looking for common characteristics of depressed people to establish an diagnosis mechanism. Lee et al. [45] investigated whether or not interviewees have depressive trends using a choice questionnaire. They recruited 93 potential depression sufferers from a Taiwan hospital and recorded patients’ condition by using two questionnaire forms, Taiwanese Depression Questionnaire (TDQ) and
Depression Scale (HADS). Finally, they used DSM-5 for reference to diagnose patients’ medical condition and the whole process took about 6 months. Park et al. [62] conducted a face-to-face interview with 14 depression sufferers to explore their daily usage of Internet. And they found there are some typical signs of depressed people surfing the Internet. For example, the 14 people are almost addicted to Internet and usually surf on net all day. These questionnaires and interviews have several limitations. For example, they are time-consuming and hard to be generalised.

Online depression detection via social media becomes possible because of the growing of social networks. Zhang et al. [87] presented social network platforms that can be regarded as a digitalization world that consists of human social experience. In many cases people would prefer to share their emotional experience and daily encounter online than talking with others offline. Marriott et al. [51] and Kulkarni et al. [41] argued that social networks users’ online personality is close to their offline personality in terms of authenticity. Moreover, the massive social networks data can support comprehensive online depression behaviours analysis. I provide an overview of online depression detection approaches in the following section.

2.2 Online depression detection based on Statistical methods

A majority of the work on identifying depression from social media attempts at making the process of diagnosing individuals automatically or semi-automatically [70, 38]. These works are primarily based on statistical methods which do not attempt to replace psychologists but aim to ease the burden of psychologists’ clinical trials. Statistical methods based approaches harvest relevant social networks users’ statistics by quantifying their online activities, social interactions and positive or negative emotion [41]. More specifically, these works collect the features of user’s online activities by analysing their posting content, counting the number of friends or followers to imply user’s social interaction and making statistics of positive or negative words number from posting texts.

Rude et al. [70] analysed the content information of depression users’ posts by
extracting typical features. They employed statistical methods to harvest information from Twitter texts, like counting the number of adjective words per tweet, counting the number of the posting. Then they asked three linguistic researchers to summarize the related topics of Twitter users’ posts. De Choudhury et al. [19] collected 476 Twitter users’ data logs of 2012 and categorised these users into depression and normal classes by referring to Center for Epidemiological Studies Depression Scale (CSE-D) questionnaires. They divided their measures of Twitter users information into three categories: (1) Social interaction measures: collect features of interacting with others, such as number of retweet, number of mention (e.g. @) others. (2) Engagement: number of posting per day and number of posting which are questions. (3) Linguistic style: usage of first person singular, adjective, noun words. Using the above measures, the authors gave some description of Twitter users’ online behaviours such as depressed users tend to use more first person singular words per tweet than normal users and normal users are more likely to mention others in their posting. Kulkarni et al. [41] employed Bag of Words (BoW) to represent Twitter texts and made statistics of high frequency words between depression and normal users to describe their online concentrations.

Moreno et al. [56] collected public Facebook profiles (e.g. number of followers, friends) from undergraduates and used DSM-5 to reference to diagnose depressed people. They described that normal users have more friends and followers than depressed users as same as daily social life. Similarly, Katikalapudi et al. [38] analysed web activities of college students and linked with depressive symptoms. They found depressed students might lack of passion in life and post some negative information online. Although these researches applied statistical techniques to extracting some features for depression detection, key steps still need manual handling such as analysing users’ posting content and classifying depressive users.

2.3 Online depression detection based on Natural Language Processing methods

Natural Language Processing (NLP) is a subfield of artificial intelligence which is used to program computers to process and analyse natural language data. The
The ultimate objective of NLP is to read, decipher, understand, and make sense of the human languages in a manner that is valuable. In online depression detection domain, the most of state of the art methods [61, 20, 74, 73] applied some NLP techniques (e.g. sentiment analysis, topic model) to extracting features from users’ posting texts because posting texts are intuitive expression of users’ daily encounters or emotional experience. Then the extracted features are used to train a machine learning classifier to form a depression detection framework.

Sentiment analysis is one of NLP techniques which is widely used to determine if social networks posting texts are positive, negative or neutral. Park et al. [61] explored the use of languages by Twitter users to describing depressive moods. In this project, they established a sentiment analysis system by using deep neural network to analyse the polarity of users’ posting text. Then the authors employed these features to train a logistic regression classifier and obtain 73% accuracy on depression users classification. They did not get a prominent classification result because their extracted features are not comprehensive. Further, Linguistic inquiry and word count (LIWC) [78] is a popular online linguistic analysis tool which can evaluate sentence polarity, cognitive processes. De Choudhury et al. [20] applied LIWC to analysing social media communication messages and inferring the changing of Twitter users’ mood during online chatting. I test the LIWC program with a self-statement depression sentence, the evaluation results are shown in Table 2.1. Based on their previous work, De Choud-

<table>
<thead>
<tr>
<th>Input Text</th>
<th>Evaluation value</th>
</tr>
</thead>
<tbody>
<tr>
<td>I am diagnosed with depression, I afraid of this disease and I might suicide.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>LIWC DIMENSION</th>
<th>Evaluation value</th>
</tr>
</thead>
<tbody>
<tr>
<td>I-WORDS (I, ME, MY)</td>
<td>21.4</td>
</tr>
<tr>
<td>SOCIAL WORDS</td>
<td>0.0</td>
</tr>
<tr>
<td>POSITIVE EMOTIONS</td>
<td>0.0</td>
</tr>
<tr>
<td>NEGATIVE EMOTIONS</td>
<td>14.3</td>
</tr>
<tr>
<td>COGNITIVE PROCESSES</td>
<td>14.3</td>
</tr>
</tbody>
</table>

Table 2.1: LIWC Evaluation Results. I-WORDS represents the percentage of first person words in whole sentence. Similarly, SOCIAL WORDS count the number of social words. POSITIVE EMOTIONS and NEGATIVE EMOTIONS compute the polarity score of input sentence, ranging from 0 to 100. COGNITIVE PROCESSES also return a value to represent sentence cognition.
hury et al. [21] introduced measures (e.g. egocentric social graphs, description of antidepressant medications and LIWC results) to quantify the online behaviors of an individual for a year before s/he reports the onset of depression. The authors utilized Principal component analysis (PCA) to reduce the dimension of feature vectors and avoid over-fitting. Finally, they employed these processed features to train a Radial Basis Function (RBF) kernel SVM and achieve a 82% accuracy on a five-fold cross validation. Their method is easy to be disturbed by noise information because they did not conduct intensive pre-processing procedures to clean data.

On the other hand, topic modeling techniques are usually employed to extract document features and analyse posting content because of its flexibility and ease of use. Shuai et al. [74] proposed a Social Network Mental Detection-base Tensor Model (STM) to detect depressed users on different social networks (e.g. Instagram, Facebook). In their project, the authors extracted commonly used features from multi-source social networks like the number of followers or friends then utilized Latent Dirichlet Allocation (LDA) topic model to mine document topics from users’ posting texts. They concatenated common features with document topic features and employed SVM as the classifier to classify them. In order to extract universal features on different social networks, they ignored some unique properties and their proposed model is not robust and stable because the employed classifier does not have a strong fitting ability on complex datasets. Nadeem et al. [57] employed a Hierarchical Dirichlet Process (HDP) to extract sentence topics and discovered that topics produced by HDP is more coherent than LDA. Moreover, they conducted experiments using four binary classifiers, such as decision trees, Naive Bayes, SVM and Logistic Regression. In their experiments, decision tree and SVM obtained best classification results separately on two Twitter datasets. These classifiers have poor ability to handle complex datasets.
2.4 State of the art online depression detection frameworks

The above methods focus on exploring new features of depression behaviours whilst ignoring the generalization ability of classifiers. Their established classifiers are linear models (e.g. SVM, LR) which cannot achieve consistent performance due to the noise or errors in the data. State of the art methods are based on both best features and classifiers.

Jamil et al. [36] proposed a user-level classifier to detect at-risk users and achieved a reasonable precision and recall. They employed BoW vectors to represent Twitter texts and collected initial features (e.g. posting time, posting count) from Twitter users data logs. Then they combined PCA with SVM to classify the extracted features. Their method obtained an accuracy of 0.6094, and F1 score of 0.6637 on CLPsych2015 dataset [17]. The whole framework is easy to implement but lacking of features from posting texts and the established classifier has poor generalization ability for complex data.

Song et al. [75] proposed a Multiple Social media Networking Learning model by seamlessly analysing information from multiple sources. They extracted features like LIWC features, user topics from multiple social network datasets (e.g. Facebook, Twitter) and presented a multi-task learning SVM to do the binary classification task. They reported linguistic features especially user topics occupy an importance position in volunteerism tendency prediction. Their proposed method provides a new idea of addressing the incomplete data problem from multiple sources but it has not been evaluated on publicly large online depression detection datasets.

Shen et al. [73] extracted six groups’ features such as user engagement and document topics with online application programming interface (API) to interpret the online behaviours of depressed users. Then they proposed a Multimodal depressive dictionary model which converting original features to sparse representations and combining these with Logistic Regression. Their proposed method obtained a state of the art result on a Twitter depression detection dataset with 7862 samples. This method harvests some special features for Twitter users classification which are benefit for improving their model performance but also limiting the universality of methods. The proposed framework has not been evaluated over different datasets.
2.4. State of the art online depression detection frameworks

Cacheda et al. [13] made detailed statistics of Twitter users posting percentages against different hours or work days. They also applied a latent semantic analysis (LSA) model to collecting text topics. The extracted features were used to train a Random Forest classifier. They reported their framework outperforms current state-of-art models by 10% on a private Twitter dataset.

However, most previous research studies focus on exploring new features of depression behaviours and the robustness of developed classifiers remains to be improved. In Chapter 3, I focus on developing a novel ensemble learning model to identify depressed users in noisy Twitter data. Firstly, I extracted several common features which can be found on multiple social networks like posting time and follower number and category them into three groups. Then, I propose a novel classifier, namely, Inverse Boosting Pruning Trees (IBPT), which demonstrates a strong classification ability on two publicly accessible Twitter depression detection datasets. The previous SOTA researches employed NLP techniques to mine text features and reported text features outperforms others in classification. But their established methods lack of interpretation ability. In Chapter 4, I propose a novel deep neural topic model, FastText Distributional Prior DocNADE with the Attention Mechanism (fdp-DocNADE) to help readers understanding what depressed users talk about on social networks and generate representative document features for depression classification. Finally, I summarize the discussed methods in Table 2.2.
<table>
<thead>
<tr>
<th>Authors</th>
<th>Critical Features</th>
<th>Employed Classifier</th>
<th>Limitations</th>
<th>Ours</th>
</tr>
</thead>
<tbody>
<tr>
<td>Park et al. [62]</td>
<td>sentence polarity, user profiles, posting time distribution...</td>
<td>(totally 7 features)</td>
<td>sentence polarity LRExtracted features are not comprehensive.</td>
<td></td>
</tr>
<tr>
<td>Nadeem et al. [57]</td>
<td>HDP topics, BoW vectors, posting time, user profiles, (totally about 20 features)</td>
<td>HDP topics,</td>
<td>Four binary classifiers, i.e. SVM, LR, Naive Bayes, Decision tree</td>
<td></td>
</tr>
<tr>
<td>Choudhury et al. [20]</td>
<td>user profiles, negative word count, LIWC, followers and friends number (totally about 35 features)</td>
<td>LIWC features, cognitive score, sentence polarity</td>
<td>SVM with RBF kernel</td>
<td></td>
</tr>
<tr>
<td>Shuai et al. [74]</td>
<td>LDA topics, user profile information (totally about 25 features)</td>
<td>LDA topics,</td>
<td>Ensemble model combine SVM and LR</td>
<td></td>
</tr>
<tr>
<td>Jamil et al. [36]</td>
<td>BoW, posting count, posting time (totally about 10 features)</td>
<td>BoW, posting count, posting time</td>
<td>BoW, posting count, posting time</td>
<td></td>
</tr>
<tr>
<td>Shen et al. [73]</td>
<td>LIWC, LSA topics, users profiles, favourite posting count (totally about 30 features)</td>
<td>LIWC, LSA topics</td>
<td>Multi-task learning + SVM</td>
<td></td>
</tr>
<tr>
<td>Shen et al. [72]</td>
<td>Six groups’ features like user engagement and domain-specific keyword (totally 35 features)</td>
<td>LDA topics,</td>
<td>Multimodal dictionary learning + LR</td>
<td></td>
</tr>
<tr>
<td>Jia et al. [13]</td>
<td>LSA topics, posting time, negative word count (totally about 15 features)</td>
<td>LSA topics</td>
<td>Random Forest</td>
<td></td>
</tr>
</tbody>
</table>

TABLE 2.2: Summary of methods for Social networks users depression classification.
Chapter 3

Inverse boosting pruning trees for depression detection on Twitter

3.1 Introduction

I have stated in the previous chapter that the established classifiers from previous researches \[73, 57, 21\] for online depression detection cannot achieve consistent performance due to the influence of noise. Ensemble learning methods have attracted much attention in the community because of its stable prediction ability. Ensemble methods use multiple learning algorithms to obtain better predictive performance than that of using any of the constituent learning algorithms alone \[34, 46, 35\]. In ensemble learning theory, multiple base classifiers (e.g. decision tree, SVM) are trained to solve the same problem and combined to get better results.

Boosting is one of popular ensemble learning methods. Boosting, that often considers homogeneous base estimators, trains estimators sequentially in an adaptive way and combines them following a deterministic strategy \[11\]. In other words, in a boosting structure, a base estimator will learn the weakness of previous ones and work out the previous errors in current training iteration. For each successive iteration, the sample weights are modified and the learning algorithm is reapplied to the reweighted data. The combined boosting model can improve its fitting ability of the training data iteratively. The prediction results of the boosting model is the major vote from all the trained base estimators. Compared with other ensemble methods (e.g. bagging, stacking), boosting can generate a combined model with lower errors as it optimises the advantages and reduces pitfalls of a single model \[26, 11, 8\].
In this work, I do not extract numerous or complicated features to achieve accurate depression detection. Instead, I develop a novel boosting classifier with a strong classification ability. My proposed framework is shown in Figure 3.1. In the first phase, I conduct data preprocessing and extract features from Twitter users (details are shown in section 3.4.2), while the second phase presents a new inverse boosting pruning trees method based on the standard Adaboost [25].

![Figure 3.1: Proposed framework.](image)

### 3.2 Review of recent boosting methods

Adaptive Boosting (Adaboost) was the first successful boosting algorithm proposed by Freund et al. [25]. More recently it may be referred to as discrete Adaboost because it is used for classification rather than regression. Here, I denote the hyperparameter tree number of Adaboost as M and N represents the training data matrix (i.e. including feature vectors and labels). In Adaboost, multiple weak learners are trained sequentially and their training error $\varepsilon_m$ is defined as follows:

$$
\varepsilon_m = \sum_{i=1}^{N} w_{m,i} d_i
$$

where $m$ is the index of a weak learner and $w_{m,i}$ represents the training weight of sample $i$, let $d_i = 1$ if the i-th case is classified incorrectly, otherwise zero. Then, samples misclassified by the preceding classifier are assigned a higher weight $w_{m+1,i}$:

$$
w_{m+1,i} = \frac{w_{m,i} \beta_i^{d_i}}{\sum_{i=1}^{N} w_{m,i} \beta_i^{d_i}}
$$
where \( \beta_m = (1 - \varepsilon_m)/\varepsilon_m \). This procedure will let subsequent classifiers pay more attention to the misclassified samples. Finally, one can combine all the weak classifiers with their weights to obtain an ensemble classifier \( G(X) = \text{sign}(\sum_{m=1}^{M} \ln(\beta_m)G_m(X)) \).

The prediction result comes from the major vote of all the weak classifiers. Algorithm 1 presents the baseline scheme of the discrete Adaboost. In general, Adaboost employs decision dump (a one-level decision tree) as its base estimator. However, decision dump cannot fit well the training data because of its simple structure. Adaboost with decision dump does not perform well in complex datasets [46].

In order to address the data fitting problem of standard Adaboost, Boonyanunta et al. [8] attempted to use different classifiers (e.g. decision dump, SVM, Naive Bayes, Logistic Regression) as the base estimators of Adaboost. In their experiment, they averaged the weight of the base estimators to reduce the variance of ensemble Adaboost models. But they found there is no obvious improvements on Adaboost with other classifiers.

Based on Adaboost, Friedman et al. [26] reported a Gradient Boosting decision trees (GBDT) which is the generalisation of boosting to arbitrary differentiable loss functions. Different from the standard Adaboost, GBDT employed regression trees as its base estimators and classification tasks will follow the regression workflow. In each iteration, a residual error is computed by the difference of the targeted value (e.g. classification labels are converted to numerical values) and the predicted value by the present estimator. Subsequent estimators will minimise the residuals as the training target. Unfortunately, GBDT can be over-fitting if the data is noisy and the training process of GBDT is time consuming.
Chapter 3. Inverse boosting pruning trees for depression detection on Twitter

Chen et al. [14] introduced an advanced Gradient Boosting algorithm (called ‘XGboost’) based on GBDT in 2016. They mainly made two contributions to optimize GBDT: (1) They grow the depth of the base decision trees to fit complex data. (2) They optimize the training speed of GBDT through assigning the training of the base trees to multiple CPUs. Although XGboost is more flexible and efficient than GBDT, it has many parameters that are hard to be tuned.

Inspired by previous works, I propose a novel algorithm to improve the performance of Adaboost in two aspects: (1) I improve the fitting and generalisation ability of the base classifier. (2) I propose a novel structure of the boosting algorithm in order to reduce the impact of the less contributing data. I introduce the proposed algorithm in Section 3.3.

3.3 Proposed Method

3.3.1 Inverse boosting pruning trees

In this section, I propose an ensemble method that combines an improved Adaboost algorithm with pruning decision trees for classification. Here, I still employ decision trees as the base estimators of boosting because of their flexibility and ease of use. Decision dump often suffers from under-fitting whilst a full tree has high variance. I here consider pruning trees in order to increase system generalisation. In my algorithm, I firstly apply all the training samples and allow a decision tree to fully grow, and then use the cost-complexity pruning method reported in [12] to prune certain branches of the trees and use the modified criterion to evaluate the system performance with the pruned trees and updated weights. Afterwards, the above steps will be executed iteratively till the maximum number of the trees is reached. To formulate my algorithm, I here declare the used notations in advance. In particular, I denote the training dataset as \( L = X_1, y_1, X_2, y_2, \ldots, X_N, y_N \). Here, \( X_n \) is the sample feature vector, \( y_n \) represents the class label and \( N \) is the number of the samples. I use \( D_m = (w_{mi}, w_{mi+1}, \ldots) \), \( m = 1, 2, \ldots, M, i = 1, 2, \ldots, N \) to represent the distribution of the sample weight in each iteration. \( M \) is the number of the estimator iterations and each sample weight is initialised to \( \frac{1}{N} \) in the first iteration during the normalisation. Furthermore, I use \( \varphi_m \) and \( G_{final}(x) \) to denote the \( m \)-th estimator’s weight and the final classifier.
Search for the best pruned tree

In the framework of Adaboost, each sample’s weight value will be updated iteratively. Therefore, I propose a weighted cost-complexity method to build the relationship between the cost-complexity value and the sample weight. Independent sample validation or cross-validation can help us to choose the best pruned subtrees. Here, I use the V-fold cross-validation for weighted cost-complexity pruning because this method can make full use of the sample information when the sample scale is not too large.

Firstly, I denote the original learning sample set $L$ which is divided randomly into $V$ subsets, $L_v, v = 1, ..., V$ and the training set of each subset is $L^{(v)} = L - L_v$. The tree $T_{\text{max}}$ comes from the original set $L$ and I create a complete tree on each subset $L_v$. I present the cost function of the decision trees as follows:

$$Gini(T, \{w\}) = \sum_{|\tilde{T}|} [Gini(\tilde{T})]$$

$$= \sum_{|\tilde{T}|} \left[ 1 - \sum_{c=1}^{C} P_c^2 \right]$$

$$= \sum_{|\tilde{T}|} \left[ 1 - \sum_{c=1}^{C} \left( \frac{\sum_{i_c} w_{m,i_c}}{\sum_{i} w_{m,i}} \right)^2 \right]$$

(3.3)

where $|\tilde{T}|$ is the leaves’ number, $C$ denotes the class number and the sample of class $c$ is defined as $i_c$. The cost of the decision trees is the sum of all the leaf nodes’ gini index. A complete tree’s gini index is zero because each leaf node only includes a single class’s samples. But the $Gini(T, \{w\})$ will increase in the pruning process where the pruned nodes’ samples are combined with their parents’ nodes. Therefore, $Gini(T, \{w\})$ is not a good measure for selecting a subtree because it always favours large trees. Thus, the penalty term, regularization parameter $\alpha$ and the tree leaves $|\tilde{T}|$ are added to the cost function. The new cost function is defined as follows:

$$R_{\alpha}(T) = Gini(T, \{w\}) + \alpha \cdot |\tilde{T}|$$

(3.4)

The penalty term favours a smaller tree when $\alpha$ is constant and $|\tilde{T}|$ decreases with pruning.

Now, the variation in the cost function is given by $R_{\alpha}(T - T_i) - R_{\alpha}(T)$, where
Chapter 3. Inverse boosting pruning trees for depression detection on Twitter

T is a complete tree, $T_i$ represents a branch with the node at $t$ and a tree pruned at node $t$ would be $T - T_i$. Next, the cost of the pruning on the internal nodes is calculated by equating $R_\alpha (T - T_i)$ to that of the branch at node $t$:

$$R_\alpha (T - T_i) - R_\alpha (T) \leq 0$$
$$\Rightarrow R_\alpha (t) - R_\alpha (T_i) \leq 0$$
$$\Rightarrow Gini(\{t\}, \{w\}) + \alpha - Gini(\{T_i\}, \{w\}) - \alpha |\tilde{T_i}| \leq 0$$  \hspace{1cm} (3.5)

$$\Rightarrow \frac{Gini(\{t\}, \{w\}) - Gini(\{T_i\}, \{w\})}{|\tilde{T_i}| - 1} \leq \alpha$$

Here I define

$$g(t) = \frac{Gini(\{t\}, \{w\}) - Gini(\{T_i\}, \{w\})}{|\tilde{T_i}| - 1}$$ \hspace{1cm} (3.6)

I will prune the branch $T_i$ with the decrease of the cost function value when $\alpha \geq g(t)$. The order of pruning is performed by setting $\alpha = \arg\min g(t)$ in order to find the branch, which should be pruned, and the process will be repeated until the tree is left with the root node only. This provides a sequence of subtrees $\{T_\alpha^{(v)}, \alpha = 0, \ldots\}$ with the associated cost-complexity parameter $\alpha$.

For $\alpha$, I apply the pruned subtree $T_\alpha^{(v)}$ to predict the estimation in the $v$-th test set, resulting in the following error rate:

$$TE_\alpha^{(v)} = \frac{\sum_{i_{miss}} w_{m,i_{miss}}^{(v)}}{\sum_{i} w_{m,i}^{(v)}}$$ \hspace{1cm} (3.7)

where $i_{miss}$ denotes the index of the misclassified sample weight, $w_{m,i}^{(v)}$ is the sample weight of the test set $L_v$ and $TE_\alpha^{(v)}$ represents the misclassified rate of set $L_v$. Hence, the average misclassified rate of $v$ is:

$$TE_\alpha = \frac{1}{v} \sum_{v=1}^{V} TE_\alpha^{(v)}$$ \hspace{1cm} (3.8)

In the meantime, I define $\alpha^* = \arg\min_\alpha TE_\alpha$, the best pruned tree obtained by pruning $T_{max}$ till $R_{\alpha^*}(T_{max})$ reaches the minimum.
3.3. Proposed Method

Inverse boosting structure

As shown in steps 4 and 5 of Algorithm 1, Adaboost employs the training error $\varepsilon_m$ as the evaluation criterion of the base estimators’ performance to set up the estimator’s weights and update the sample weights. But $\varepsilon_m$ is not a consistent criterion against the pruned trees. Figure 3.2 illustrates the error rate of the training and validation set corresponding to the pruning. I observe that the training samples’ errors $\varepsilon_m$ keep increasing during the pruning, and the validation samples’ errors $\varepsilon^s_m$ decrease when the tree is relatively small. The validation samples’ errors come to the minimum with the subtree of approximate 30 leaf nodes, and begin to increase when the tree continues to be pruned. $\varepsilon_m$ is not a suitable criterion for evaluating the pruned trees. In fact, the pruned trees should be given higher estimation weights when they have a lower validation error. Thus, I now propose a novel boosting structure which associates the classification outcome with the pruned trees accordingly.

In the first step, I fit a complete decision tree to the training data $L$ and prune it in order to obtain the best tree structure. Next, I use $TE_\alpha$ to replace $\varepsilon_m$ as the evaluation criterion because $TE_\alpha$ closely reflects the pruned trees’ outcome. The estimator’s weight can be updated as follows,

$$
\varphi_m = \ln \frac{1 - TE_\alpha}{TE_\alpha} \quad \text{s.t.} \quad TE_\alpha < \frac{1}{2} \quad (3.9)
$$

when $TE_\alpha \geq \frac{1}{2}$, I allow the iteration process to terminate a little bit early because the current estimator cannot maintain the classification performance at all times.

Figure 3.2: The error rate curve of the training and validation sets corresponding to the pruning process.
In the next step, I present an inverse boosting structure to update the sample weight. In each iteration, I treat the misclassified samples as ‘disturbing’ items which may cloud the judgement of the estimator from the pruned branch and influence the generalisation ability of the decision trees. Hence, the sample weight can be updated as follows:

\[
\begin{align*}
    w_{m+1,i} &= \frac{w_m i \exp(\phi_m I(G_m(X_i), y_i))}{Z_m}, \quad i = 1, 2, \ldots, N \\
    D_{m+1} &= (w_{m+1,1}, w_{m+1,2}, \ldots, w_{m+1,N}), \quad m = 1, 2, \ldots, M
\end{align*}
\]

In Eqs. (3.10), the normalisation factor \(Z_m\) is

\[
Z_m = \sum_i w_m i \exp(\phi_m I(G_m(X_i), y_i))
\]

\(G_m(X_i)\) is the predicted value of \(X_i\) by the pruned tree \(G_m\), \(y_i\) represents the ground-truth of sample \(X_i\), and \(I(G_m(X_i), y_i)\) is defined as:

\[
I(G_m(X_i), y_i) = \begin{cases} 
1, & G_m(X_i) = y_i \\
-1, & G_m(X_i) \neq y_i
\end{cases}
\]

The misclassified samples’ weights are decreasing and the weight of the classified samples are increasing, which reduces the influence of the ‘disturbing’ items for reforming the trees in a later iteration. Then, I use the dataset \(L\) with the updated weight distribution \(D_{m+1}\) to train the next estimator, set the estimator weight and update the sample weight iteratively in order to satisfy the maximum number of the estimators. The final ensemble classifier will be:

\[
G_{final}(X) = \text{sign}(\sum_{m=1}^M \phi_m^m G_m(X))
\]

The proposed algorithm is based on inverse boosting pruning trees, which is illustrated in Algorithm 2. I discuss the convergence characteristics of the proposed classification approach in the following section.

### 3.3.2 Convergence Analysis

Convergence is a descriptive characteristic of a classifier. In this section, I look closely at the convergence of my proposed classification method, which shows
**Algorithm 2 Inverse boosting pruning trees based algorithm.**

**Require:** $M$ - Trees’ number, $N$ - Samples number, $L$ - Learning samples, and $V$ - Folds.

1: **function** BESTPRUNEDSUBTREE($L, V, D_m$)
2: Split the learning samples $L$ into $V$ folds, $L_v, v = 1, 2, ..., V$, and grow a max tree $T_{max}$ on $L$.
3: Test sample set $L^v = L - L_v$.
4: **for** $v \in [1, V]$ **do**
5: Fit a decision tree to $L_v$ training samples.
6: Subtree sequence $\{T_{\alpha}^{(v)}, \alpha = 0, \ldots\} \leftarrow R_{\alpha} (T - T_t) - R_{\alpha} (T) \leq 0$ ▷ Recursively repeat till the tree only has root nodes: 1. Calculate $g(t)$ using Eq. (3.6). 2. Set $\alpha = \text{argmin} g(t)$ and prune the branch $T_t$.
7: Calculate $TE_{\alpha}^{(v)}$ by Eq. (3.7).
8: **end for**
9: Compute $TE_{\alpha} \leftarrow \frac{1}{V} \sum_{v=1}^{V} TE_{\alpha}^{(v)}$.
10: Define $\alpha^* \leftarrow \text{argmin}_{\alpha} TE_{\alpha}$.
11: The best pruned tree $G_m(X)$ is obtained by pruning $T_{max}$ till $R_{\alpha^*}(T_{max})$ becomes minimal.
12: **return** $G_m(X), TE_{\alpha}$.
13: **end function**

15: **function** INVERSE BOOSTING($L, V, M, N$)
16: Initialise sample weight distribution $D_m = (w_{mi})..., m = 1, 2, ..., M, i = 1, 2, ..., N$ and set each sample weight $w_{mi}$ to $\frac{1}{N}$.
17: **for** $m \in (1, M)$ **do**
18: $G_m(X), TE_{\alpha} \leftarrow$ BestPrunedSubtree ($L, V, D_m$).
19: Update the estimator weight using Eq. (3.9).
20: Update each sample’s weight $w_{mi}$ using Eqs. (3.10) and (3.11).
21: Preserve $D_{m+1}$ for the next iteration.
22: **end for**
23: **return** Final ensemble classifier $G_{\text{final}}(X) \leftarrow \text{sign}(\sum_{m=1}^{M} W_i^m G_m(X))$.
24: **end function**
that the pruning process is model adaptive, stating in particular that the convergence error of the associated classification rule is strictly monotone decreasing. The analysis also indicates that the rate of convergence can be maintained with adding random noise to the data or different estimator numbers.

Firstly, Eq. (3.10) can be transformed as follows:

\[ w_{m+1,i} = \frac{w_m \exp(\phi_m I(G_m(X_i), y_i))}{Z_m} = w_{1,i} \cdot \frac{\exp(\phi_1 I(G_1(X_i), y_i))}{Z_1} \cdots \frac{\exp(\phi_m I(G_m(X_i), y_i))}{Z_m} \]

\[ = \frac{1}{N} \cdot \frac{\exp(\sum_m \phi_m I(G_m(X_i), y_i))}{\prod_m Z_m} \]

\[ = \frac{1}{N} \cdot \frac{\exp(y_i G_{\text{final}}(X_i))}{\prod_m Z_m} \]

Thereby, the training error of the final classifier \( \varepsilon_{\text{final}} \) is shown as follows:

\[ \varepsilon_{\text{final}} = \frac{1}{N} \sum_i \left\{ \begin{array}{ll} 1 \text{ s.t. } y_i \neq G_{\text{final}}(X_i) & \\ 0 \text{ s.t. } y_i = G_{\text{final}}(X_i) & \end{array} \right. \]

\[ = \frac{1}{N} \sum_i \left\{ \begin{array}{ll} 1 \text{ s.t. } y_i G_{\text{final}}(X_i) \leq 0 & \\ 0 \text{ s.t. } y_i G_{\text{final}}(X_i) > 0 & \end{array} \right. \]

when \( y_i \neq G_{\text{final}}(X_i) \), \( \varepsilon_{\text{final}} \leq \frac{1}{N} \cdot \sum_i \exp(-y_i G_{\text{final}}(X_i)) \) because \( \exp(-t) \geq 1 \) if \( t \leq 0 \). Then, I have the following equation:

\[ N \cdot \exp(-y_i G_{\text{final}}(X_i)) = \frac{1}{w_{m+1,i}} \cdot \frac{1}{\prod_m Z_m} \]

Finally, Eq. (3.16) can be represented as:

\[ \varepsilon_{\text{final}} \leq \frac{1}{N} \cdot \exp(-y_i G_{\text{final}}(X_i)) \]

\[ \leq \frac{1}{N^2} \cdot \frac{1}{\prod_m Z_m} \cdot \sum_i \frac{1}{w_{m+1,i}} \]

where \( w_{m+1,i} \) are the updated weights and \( \sum_i w_{m+1,i} = 1 \) because of the normalisation. According to Cauchy–Schwarz inequality, I have:

\[ \sum_i w_{m+1,i} \cdot \sum_i \frac{1}{w_{m+1,i}} \geq N^2 \]
The upper limit of the training error $\varepsilon_{\text{final}}$ can be described as:

$$\varepsilon_{\text{final}} \leq \frac{1}{\prod_{m} Z_{m}}$$  \hspace{1cm} (3.20)

It is observed that the upper limit of the training error $\varepsilon_{\text{final}}$ is strictly monotone decreasing when $Z_{m} > 1$. As I have discussed above, $\sum_{i} w_{m,i} = 1$. Assuming $Z_{m} > 1$, I have:

$$|w_{r} \cdot \exp(\varphi_{m}) - w_{r}| > |w_{w} \cdot \exp(-\varphi_{m}) - w_{w}|$$

$$\Rightarrow w_{r} \cdot \exp(\varphi_{m}) - w_{r} > w_{w} - w_{w} \cdot \exp(-\varphi_{m})$$

$$\Rightarrow \exp(\varphi_{m}) \cdot \frac{1}{1 - \exp(-\varphi_{m})} > \frac{w_{w}}{w_{r}}$$

$$\Rightarrow \exp(\varphi_{m}) > \frac{w_{w}}{w_{r}}$$  \hspace{1cm} (3.21)

where $w_{r}$ represents the weight of the classified sample and $w_{w}$ is the misclassified sample weight. When Eq. (3.21) is satisfied, I ensure $Z_{m} > 1$. Because of my inverse boosting structure, the base estimator error must be less than 0.5 and the misclassified samples’ weights are decreasing iteratively, where I hold $\frac{w_{w}}{w_{r}} \leq 1$. According to the above definition, I have $TE_{\alpha} < \frac{1}{2}$, and $\exp(\varphi_{m}) > 1$. Thus, Eq. (3.21) always holds, and I can verify $Z_{m} > 1$. This suggests that the training error of my proposed method is strictly monotone decreasing with the increasing estimator number $m$ and eventually the errors converges to a global minimum.

### 3.4 Experiment Setup

I set up experiments to evaluate the classification performance of IBPT on Two Twitter depression datasets, then compare the results with 12 baselines.

#### 3.4.1 Datasets

I conduct experiments on two public Twitter depressed users datasets to verify the effectiveness of my proposed novel framework. The details of experimental datasets are introduced as follows.

**Tsinghua Twitter Depression Database (TTDD):** The Twitter database collected by Shen et al.[73] in 2017 for depression detection. The Twitter database
Chapter 3. Inverse boosting pruning trees for depression detection on Twitter

has three parts: (1) **Depression Dataset D1**: The dataset was created based on the tweets collected between 2009 and 2016, where the users were labelled as depressed people if their anchor tweets satisfied the pattern "(I’m/I was/I am/I’ve been) diagnosed depression". (2) **Depression Dataset D2**: This dataset collects Twitter messages that users were labelled as non-depressed if they had never posted any tweets containing the character string "depress". (3) **Depression Dataset D3**: Shen et al. [73] constructed an unlabelled large dataset D3 for depression candidates. Based on the tweets shown in December 2016, this unlabelled depression candidates dataset collected users if their anchor tweets loosely contained the character string "depress". There are 2558, 5304 and 58810 samples stored in D1, D2, D3 respectively. Each sample of these three datasets contains one-month post information of a Twitter user before the anchor tweet was detected. In this Chapter, I employ the well labelled datasets D1 and D2 to evaluate my classification algorithm’s performance and analyse the online behaviours of depressed users.

**CLPsych 2015 Twitter Dataset**: The dataset was established by John Hopkins University for a depression detection share task in 2015. The dataset collected public Twitter users’ posts between 2008 and 2013 via Twitter application programming interface (API). Similarly, possible mental disease sufferers was labeled as depression or PTSD according to their self statement of diagnosis, like ‘I was just diagnosed with depression or PTSD...’. Furthermore, they conducted careful pre-preprocessing procedures and anonymization operations, such as filtering users whose tweets are fewer than 25 and removing individual information. Finally, they manually examined and refined the annotation of each collected Twitter user’s logs by using a semi-supervised method. The processed dataset consisted of 477 depressed users, 396 PTSD (i.e. Post-traumatic stress disorder (PTSD) is an anxiety disorder caused by very stressful, frightening or distressing events) users and 873 control users. For each user, up to their most recent 3000 public tweets were included in the dataset.

All experimental procedures have been approved by University of Leicester Ethical Review body (shown in Appendix A). I summarise the statistics of above two depression datasets in Table 3.1.
3.4. Experiment Setup

### Dataset

<table>
<thead>
<tr>
<th></th>
<th>Depressed Users</th>
<th>PTSD Users</th>
<th>Normal Users</th>
<th>Class Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>TTDD</td>
<td>2.5k</td>
<td>~</td>
<td>5.3k</td>
<td>Depression, normal</td>
</tr>
<tr>
<td>CLPsych 2015</td>
<td>477</td>
<td>396</td>
<td>873</td>
<td>Depression, PTSD, Control</td>
</tr>
</tbody>
</table>

**Table 3.1:** Twitter depression dataset Statistics.

#### 3.4.2 Data Preprocessing and feature extraction

As the scripts on Twitter may be random and unpredictable, features with different noise may be obtained and influence the detection accuracy. Before feature extraction is implemented, I carry out the following preprocessing procedure: (1) Minimisation of the influence of noisy samples. Inspired by the work of Yazdavar et al. [84], I remove samples from the dataset where the posting number of the samples is less than five. These samples cannot provide sufficient information for analysing the users' behaviours or topic modelling. (2) Processing of irregular words. The words on social networks may look irregular because of mistaken spelling or abbreviations. I use the Textblob API reported in [50] (commonly used in natural language processing tasks) to remedy the wrong type of words. (3) Stemming. I expect to perform statistical analysis on commonly used words of normal and depressed users separately and conduct topic modelling on the users' posts. Words must be of unified representations regardless of tense and voice. Hence, I utilise the SnowballStemmer algorithm reported in [66] to deal with these words. For instance, “accepting” and “accepted” can be converted to “accept”. Afterwards, I extract three feature categories as follows and the proposed framework is shown in Phase 1 of Figure 3.1.

1. User’s Profile Features: The user’s profile features contain individual information on social networks. I collect 4 different features here: total_favourites reflects the number of posts that this particular user favours during his/her account’s lifetime; listed_count shows the number of the public list that this user holds a membership within. I collect the number of the user’s friends and followers which well characterise the author’s egocentric social networks.

2. Social Interaction Features: Park et al. [61] discovered that depressed users are less active in social networks, and depressed users regard social networking as a tool for social awareness and emotional interaction. Thus, I extract retweet count, mention count (e.g. @someone) and favourites count (indicating
how many times this post has been favoured by other users) to describe the behaviours of the user interacting with others. Besides, I collect the posting number and time distribution to demonstrate the user’s activeness on social networks.

(3) Linguistic Features: The content of the posts on social networks can intuitively reflect a person’s mood and attitude. Depressed users may post more negative words than normal users [18, 61, 21, 73]. Hence, I count the numbers of negative and positive words in the tweets using the NLTK toolkit [6]. In addition, I collect the numbers of emojis and emoticons from the texts to form relevant features. In order to comprehensively explore the semantics, Resnik et al. [68] examined the difference of the concerned topics between the depression and normal users by topic models and observed that topic models might be effective for depression detection. In my work, I utilise the Latent Dirichlet Allocation (LDA) approach presented in [7] to extract topic distributions from the tweets.

3.4.3 Baseline Comparison Methods

Decision tree: A decision tree based classifier partitions data recursively to form groups or classes. This is a supervised learning algorithm which can be used in discrete or continuous data for classification or regression [71].

Random Forest: Random Forest is an ensemble learning method of using bagging and random features selection to construct a multitude of decision trees during the training [12, 39]. This classification algorithm is widely used in data mining area.

Discrete Adaboost: I have introduced this algorithm in Section 3.2. I abbreviate Discrete Adaboost algorithm as Dis_Ada in the next experiment section.

Real Adaboost: Real Adaboost is an extension of the classical Adaboost algorithm. Real AdaBoost is able to produce readable credit scorecards and offers attractive features including variable interaction and adaptive, stage-wise binning [23]. I abbreviate Real Adaboost algorithm as Real_Ada in the next experiment section.

LogitBoost: LogitBoost algorithm is derived by considering AdaBoost as a generalised additive model whilst applying the cost functional of logistic regression.

SVM: Support Vector Machine (SVM) is a supervised learning model with an associated learning algorithm that analyses data for classification or regression.
3.5. Experimental Result

Because of the kernel tricks, SVM cannot only perform linear classification, but also perform efficiently on non-linear classification.

**Naive Bayes**: Naive Bayes classifiers are a family of simple "probabilistic classifiers" based on Bayes’ theorem with strong (naive) independence assumptions between the features.

**XGboost**: XGBoost is an optimised distributed gradient boosting library designed to be highly efficient, flexible and portable. It implements machine learning algorithms under the Gradient Boosting framework. XGBoost provides a parallel tree boosting (also known as GBDT, GBM) that solves the classification problems in a fast and accurate way [14].

**GBDT**: The idea of gradient boosting decision trees (GBDT) proposed by Breiman et al. [10] is that boosting can be interpreted as an optimization algorithm on a suitable cost function. GBDT builds an additive model in a forward stage-wise fashion and allows for the optimization of arbitrary differentiable loss functions.

**Adaboost with Naive Bayes**: I employ Naive Bayes as the base estimators of discrete boosting structure and evaluate the performance of this ensemble algorithm. For convenience, I denote Adaboost with Naive Bayes as Ada_NB.

**Adaboost with support vector machine**: I employ SVM as the base estimators of discrete boosting structure and test the performance of this ensemble algorithm. For convenience, I denote Adaboost with SVM as Ada_SVM.

**Adaboost with pruning trees**: I combine the Discrete Adaboost with pruning trees as a comparison classification method against IBPT to verify the effectiveness of my proposed inverse boosting structure. Adaboost with pruning trees is abbreviated as Ada_PT in the following section.

### 3.5 Experimental Result

In this section, I analyse the correlation between the parameters of the IBPT firstly. Then I conduct experiment to verify the IBPT’s classification ability in three toy datasets in comparison with the results of the aforementioned 12 methods. Finally, I evaluate the performance of the IBPT for depression detection on two publicly accessible Twitter datasets and explore the online behaviours of the depressed users on Twitter.
3.5.1 IBPT parameter tuning

![Figure 3.3](image)

**Figure 3.3:** This figure shows the IBPT’s error rate, estimator error, estimator weight changing with the number of the estimators against the depression dataset: (a) Test error curve in training. (b) Estimator error curve. (c) Estimator weight curve.

I conduct experiments on the depression dataset to evaluate the performance of my proposed algorithm IBPT in practice. My approach only has two parameters, base estimator number \( M \) and pruning cross-validation fold \( V \). Figure 3.3(a) shows the result of the IBPT with different pruning folds including training and test errors with the number of estimators. For comparison, I include the results of boosting based algorithms in the same figure. It is observed that with the increment of the estimators’ number, all the methods’ training and test errors keep decreasing and converge to a minimum value, which confirms the lower limit of the IBPT. Moreover, IBPT’s training errors converge to zero and the IBPT’s final testing errors are less than those of other boosting algorithms, which verifies the improvement of my algorithm on the generalisation ability over Adaboost. Obviously, there is no significant performance changes of the IBPT with different pruning folds \( V \).

Meanwhile, I also include the base estimators’ errors, and the weight variation curve is shown in Figure 3.3(b) and 3.3(c). Here, I include the Discrete and Real Adaboost results in Figure 3.3(b) and 3.3(c) for comparison. Discrete Adaboost’s estimation errors increase with the increasing of the estimators’ number and are inversely proportional to the estimators’ weights. Real Adaboost does not have the section of updating the estimators’ weight so its estimators’ weights remain unchanged, but my proposed IBPT’s estimation errors or weights vary in
a certain interval. This is due to the fact that the pruning trees increase the fitting ability against the training data and the inverse boosting structure reduces the estimators’ errors through continuously decreasing misclassified sample weights.

To summarise, I have verified the convergence of the IBPT and comprehensively analysed the variation of the IBPT compared with the traditional Adaboost. Estimators’ number $M$ is the crucial parameter of the IBPT, and I use 500 base estimators and set the pruning fold $V$ to 10 in order to ensure that the IBPT achieves consistent performance in the following section.

### 3.5.2 Toy experiments

In order to evaluate my method comprehensively, I use three real datasets from the UCI machine learning repository [5] to examine my method’s classification performance and compare my method with other 12 baselines. The testing datasets are chosen as follows: (1) **LSVT Dataset**. This dataset collected clinical information from speech signals and contained 128 samples, 309 features and 2 classes [79]. (2) **News Dataset**. This dataset collected information of the popularity of online news and included 39797 samples, 58 features, and 2 classes [24]. (3) **Glass Dataset**. This dataset collected information of glasses left in crime scenes and contained 214 samples, 10 features, and 6 classes [9]. I adopt the Area Under the curve of ROC (AUC) as the evaluation criteria for the algorithm classification performance. Moreover, in order to exclude experimental contingency and make full use of sample information, I employ a 5-folds cross validation to conduct the experiments.

The results of classification on three datasets are presented in Tables 3.4, 3.5 and 3.6. I have obtained the following observations: (1) My proposed method Inverse Boosting Pruning trees (IBPT) achieves the best performance for the three datasets. (2) Combining the boosting structure with the decision trees can improve the classification performance, compared with the other boosting ensemble methods such as Ada_NB, Ada_SVM and this demonstrates the value of the improved decision tree structure. (3) It is evident that the ensemble methods such as Ada_PT, XGboost, GBDT, Random Forest outperform the other traditional machine learning classifiers (e.g. SVM, Naive Bayes) for the three datasets, which confirm the ensemble learning methods have a strong classification ability. (4) On the LSVT dataset, IBPT’s AUC score surpasses that of Discrete Adaboost and
Chapter 3. Inverse boosting pruning trees for depression detection on Twitter

<table>
<thead>
<tr>
<th>Method</th>
<th>1st-fold</th>
<th>2nd-fold</th>
<th>3rd-fold</th>
<th>4th-fold</th>
<th>5th-fold</th>
<th>Mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proposed Method</td>
<td>0.9124</td>
<td>0.9142</td>
<td>0.9117</td>
<td>0.9263</td>
<td>0.9167</td>
<td>0.9154</td>
</tr>
<tr>
<td>Decision tree</td>
<td>0.7290</td>
<td>0.7322</td>
<td>0.7311</td>
<td>0.7228</td>
<td>0.7597</td>
<td>0.7347</td>
</tr>
<tr>
<td>Random Forest</td>
<td>0.9033</td>
<td>0.9085</td>
<td>0.9062</td>
<td>0.9118</td>
<td>0.9086</td>
<td>0.9070</td>
</tr>
<tr>
<td>Real_Ada</td>
<td>0.8847</td>
<td>0.8844</td>
<td>0.8848</td>
<td>0.8848</td>
<td>0.8932</td>
<td>0.8861</td>
</tr>
<tr>
<td>Dis_Ada</td>
<td>0.9000</td>
<td>0.8934</td>
<td>0.9000</td>
<td>0.8880</td>
<td>0.8884</td>
<td>0.8939</td>
</tr>
<tr>
<td>LogitBoost</td>
<td>0.9079</td>
<td>0.9081</td>
<td>0.9125</td>
<td>0.9059</td>
<td>0.9036</td>
<td>0.9076</td>
</tr>
<tr>
<td>SVM</td>
<td>0.5379</td>
<td>0.5489</td>
<td>0.5318</td>
<td>0.5329</td>
<td>0.5309</td>
<td>0.5364</td>
</tr>
<tr>
<td>Naive Bayes</td>
<td>0.7714</td>
<td>0.7624</td>
<td>0.7783</td>
<td>0.7962</td>
<td>0.8147</td>
<td>0.7847</td>
</tr>
<tr>
<td>XGboost</td>
<td>0.8997</td>
<td>0.9003</td>
<td>0.8977</td>
<td>0.9026</td>
<td>0.8994</td>
<td>0.8994</td>
</tr>
<tr>
<td>GBDT</td>
<td>0.8873</td>
<td>0.8914</td>
<td>0.8892</td>
<td>0.8939</td>
<td>0.8886</td>
<td>0.8897</td>
</tr>
<tr>
<td>Ada_NB</td>
<td>0.4888</td>
<td>0.6302</td>
<td>0.5537</td>
<td>0.5600</td>
<td>0.6346</td>
<td>0.5734</td>
</tr>
<tr>
<td>Ada_SVM</td>
<td>0.5511</td>
<td>0.5480</td>
<td>0.5178</td>
<td>0.5220</td>
<td>0.5241</td>
<td>0.5325</td>
</tr>
<tr>
<td>Ada_PT</td>
<td>0.9124</td>
<td>0.9132</td>
<td>0.8887</td>
<td>0.8928</td>
<td>0.9078</td>
<td>0.9029</td>
</tr>
</tbody>
</table>

Table 3.2: TTDD Dataset. The evaluation criterion is Area Under the curve of ROC (AUC) score.

<table>
<thead>
<tr>
<th>Method</th>
<th>1st-fold</th>
<th>2nd-fold</th>
<th>3rd-fold</th>
<th>4th-fold</th>
<th>5th-fold</th>
<th>Mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proposed Method</td>
<td>0.8381</td>
<td>0.8260</td>
<td>0.8006</td>
<td>0.8063</td>
<td>0.8002</td>
<td>0.8142</td>
</tr>
<tr>
<td>Decision tree</td>
<td>0.6358</td>
<td>0.6401</td>
<td>0.6477</td>
<td>0.6563</td>
<td>0.6195</td>
<td>0.6399</td>
</tr>
<tr>
<td>Random Forest</td>
<td>0.8299</td>
<td>0.8198</td>
<td>0.8044</td>
<td>0.7952</td>
<td>0.7930</td>
<td>0.8083</td>
</tr>
<tr>
<td>Real_Ada</td>
<td>0.7493</td>
<td>0.7005</td>
<td>0.7127</td>
<td>0.6775</td>
<td>0.6963</td>
<td>0.7074</td>
</tr>
<tr>
<td>Dis_Ada</td>
<td>0.7743</td>
<td>0.7230</td>
<td>0.7296</td>
<td>0.7070</td>
<td>0.7280</td>
<td>0.7325</td>
</tr>
<tr>
<td>LogitBoost</td>
<td>0.7952</td>
<td>0.7808</td>
<td>0.7611</td>
<td>0.7348</td>
<td>0.7599</td>
<td>0.7665</td>
</tr>
<tr>
<td>SVM</td>
<td>0.6479</td>
<td>0.6376</td>
<td>0.6416</td>
<td>0.6343</td>
<td>0.6139</td>
<td>0.6128</td>
</tr>
<tr>
<td>Naive Bayes</td>
<td>0.5718</td>
<td>0.6181</td>
<td>0.5907</td>
<td>0.6934</td>
<td>0.5897</td>
<td>0.6128</td>
</tr>
<tr>
<td>XGboost</td>
<td>0.8306</td>
<td>0.8219</td>
<td>0.7995</td>
<td>0.7893</td>
<td>0.8081</td>
<td>0.8098</td>
</tr>
<tr>
<td>GBDT</td>
<td>0.8321</td>
<td>0.8175</td>
<td>0.7972</td>
<td>0.7784</td>
<td>0.8161</td>
<td>0.8084</td>
</tr>
<tr>
<td>Ada_NB</td>
<td>0.5654</td>
<td>0.5086</td>
<td>0.5297</td>
<td>0.5716</td>
<td>0.5943</td>
<td>0.5540</td>
</tr>
<tr>
<td>Ada_SVM</td>
<td>0.6468</td>
<td>0.6429</td>
<td>0.6424</td>
<td>0.6181</td>
<td>0.6135</td>
<td>0.6368</td>
</tr>
<tr>
<td>Ada_PT</td>
<td>0.8261</td>
<td>0.8119</td>
<td>0.7998</td>
<td>0.7783</td>
<td>0.7941</td>
<td>0.8020</td>
</tr>
</tbody>
</table>

Table 3.3: The CLPsych Dataset. The evaluation criterion is Area Under the curve of ROC (AUC) score.
3.5. Experimental Result

<table>
<thead>
<tr>
<th></th>
<th>1st-fold</th>
<th>2nd-fold</th>
<th>3rd-fold</th>
<th>4th-fold</th>
<th>5th-fold</th>
<th>Mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proposed Method</td>
<td>0.8833</td>
<td>0.8377</td>
<td>0.8859</td>
<td>0.9481</td>
<td>0.9265</td>
<td>0.8948</td>
</tr>
<tr>
<td>Decision tree [71]</td>
<td>0.4500</td>
<td>0.7013</td>
<td>0.5044</td>
<td>0.7013</td>
<td>0.6948</td>
<td>0.6104</td>
</tr>
<tr>
<td>Random Forest [12]</td>
<td>0.8833</td>
<td>0.8377</td>
<td>0.8596</td>
<td>0.9058</td>
<td>0.8823</td>
<td>0.8731</td>
</tr>
<tr>
<td>Real_Ada [23]</td>
<td>0.8000</td>
<td>0.8117</td>
<td>0.8246</td>
<td>0.8571</td>
<td>0.8897</td>
<td>0.8364</td>
</tr>
<tr>
<td>Dis_Ada [25]</td>
<td>0.7666</td>
<td>0.8052</td>
<td>0.8421</td>
<td>0.8636</td>
<td>0.8603</td>
<td>0.8275</td>
</tr>
<tr>
<td>LogitBoost [40]</td>
<td>0.7125</td>
<td>0.8539</td>
<td>0.8246</td>
<td>0.8052</td>
<td>0.8419</td>
<td>0.8076</td>
</tr>
<tr>
<td>SVM [31]</td>
<td>0.5000</td>
<td>0.5000</td>
<td>0.5000</td>
<td>0.5000</td>
<td>0.5000</td>
<td>0.5000</td>
</tr>
<tr>
<td>Naive Bayes [86]</td>
<td>0.7667</td>
<td>0.7078</td>
<td>0.6667</td>
<td>0.7272</td>
<td>0.5514</td>
<td>0.6853</td>
</tr>
<tr>
<td>XGboost [14]</td>
<td>0.7917</td>
<td>0.7857</td>
<td>0.8596</td>
<td>0.9026</td>
<td>0.8602</td>
<td>0.8399</td>
</tr>
<tr>
<td>GBDT [10]</td>
<td>0.7500</td>
<td>0.8311</td>
<td>0.8070</td>
<td>0.9350</td>
<td>0.7794</td>
<td>0.8206</td>
</tr>
<tr>
<td>Ada_NB</td>
<td>0.6167</td>
<td>0.5682</td>
<td>0.6929</td>
<td>0.7727</td>
<td>0.5514</td>
<td>0.6410</td>
</tr>
<tr>
<td>Ada_SVM</td>
<td>0.5000</td>
<td>0.5000</td>
<td>0.5000</td>
<td>0.5000</td>
<td>0.5000</td>
<td>0.5000</td>
</tr>
<tr>
<td>Ada_PT</td>
<td>0.8033</td>
<td>0.8116</td>
<td>0.8684</td>
<td>0.9221</td>
<td>0.6948</td>
<td>0.8200</td>
</tr>
</tbody>
</table>

Table 3.4: LSVT Dataset. The evaluation criterion is Area Under the curve of ROC (AUC) score.

Ada_PT by 6% and 7% respectively, and the IBPT outperforms these two methods in News and Glass datasets, which verifies the inverse boosting structure and pruning trees effectively work together to achieve better system performance. Besides that, I generate the statistics of feature importance of the IBPT for three toy datasets (shown in Figure 3.4(a)-3.4(c)) to verify the IBPT’s effectiveness for distinguishing features and the reliability of the classification results. I extract two important features in each dataset and then plot Figure 3.5(a)-(f) to introduce the difference between the classes in each dataset. It is observed that there is significant difference of sample distributions between classes in the more important features shown in Figure 3.5(a), (c) and (e). But in the less important features shown in Figure 3.5(b), (d) and (f), different classes’ sample distributions are overlapped each other so it is difficult to separate different classes in these cases. In summary, it shows that the IBPT is effective for distinguishing features and its classification result is reliable.

Through the experiments of classification on the three toy datasets, my proposed method IBPT shows a strong classification ability in complex datasets with multi-features or multi-classes. In the next section, I employ the IBPT for depression detection and analyse depression features in order to understand the online behaviours of the depressed users.
34 Chapter 3. Inverse boosting pruning trees for depression detection on Twitter

<table>
<thead>
<tr>
<th>Method</th>
<th>1st-fold</th>
<th>2nd-fold</th>
<th>3rd-fold</th>
<th>4th-fold</th>
<th>5th-fold</th>
<th>Mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proposed Method</td>
<td>0.7809</td>
<td>0.7896</td>
<td>0.7863</td>
<td>0.7773</td>
<td>0.7820</td>
<td>0.7830</td>
</tr>
<tr>
<td>Decision tree [71]</td>
<td>0.5818</td>
<td>0.5995</td>
<td>0.5917</td>
<td>0.6003</td>
<td>0.6045</td>
<td>0.5955</td>
</tr>
<tr>
<td>Random Forest [12]</td>
<td>0.7612</td>
<td>0.7620</td>
<td>0.7631</td>
<td>0.7521</td>
<td>0.7468</td>
<td>0.7569</td>
</tr>
<tr>
<td>Real_Ada [23]</td>
<td>0.7598</td>
<td>0.7531</td>
<td>0.7600</td>
<td>0.7499</td>
<td>0.7612</td>
<td>0.7567</td>
</tr>
<tr>
<td>Dis_Ada [25]</td>
<td>0.7152</td>
<td>0.7156</td>
<td>0.7051</td>
<td>0.7099</td>
<td>0.7155</td>
<td>0.7123</td>
</tr>
<tr>
<td>LogitBoost [40]</td>
<td>0.7408</td>
<td>0.7344</td>
<td>0.7239</td>
<td>0.7314</td>
<td>0.7309</td>
<td>0.7323</td>
</tr>
<tr>
<td>SVM [31]</td>
<td>0.5013</td>
<td>0.5035</td>
<td>0.5021</td>
<td>0.5012</td>
<td>0.5025</td>
<td>0.5019</td>
</tr>
<tr>
<td>Naive Bayes [86]</td>
<td>0.5165</td>
<td>0.5382</td>
<td>0.5286</td>
<td>0.5206</td>
<td>0.5302</td>
<td>0.5268</td>
</tr>
<tr>
<td>XGboost [14]</td>
<td>0.7627</td>
<td>0.7703</td>
<td>0.7635</td>
<td>0.7450</td>
<td>0.7580</td>
<td>0.7607</td>
</tr>
<tr>
<td>GBDT [10]</td>
<td>0.7657</td>
<td>0.7678</td>
<td>0.7599</td>
<td>0.7488</td>
<td>0.7542</td>
<td>0.7591</td>
</tr>
<tr>
<td>Ada_NB</td>
<td>0.4982</td>
<td>0.4929</td>
<td>0.4951</td>
<td>0.4947</td>
<td>0.5069</td>
<td>0.4976</td>
</tr>
<tr>
<td>Ada_SVM</td>
<td>0.5012</td>
<td>0.5027</td>
<td>0.5020</td>
<td>0.5005</td>
<td>0.5013</td>
<td>0.5015</td>
</tr>
<tr>
<td>Ada_PT</td>
<td>0.7293</td>
<td>0.7193</td>
<td>0.7116</td>
<td>0.7224</td>
<td>0.7234</td>
<td>0.7212</td>
</tr>
</tbody>
</table>

Table 3.5: News Dataset. The evaluation criterion is Area Under the curve of ROC (AUC) score.

<table>
<thead>
<tr>
<th>Method</th>
<th>1st-fold</th>
<th>2nd-fold</th>
<th>3rd-fold</th>
<th>4th-fold</th>
<th>5th-fold</th>
<th>Mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proposed Method</td>
<td>0.9140</td>
<td>0.9644</td>
<td>0.9720</td>
<td>0.9599</td>
<td>0.9866</td>
<td>0.9585</td>
</tr>
<tr>
<td>Decision tree [71]</td>
<td>0.8047</td>
<td>0.7767</td>
<td>0.8744</td>
<td>0.8325</td>
<td>0.8857</td>
<td>0.8346</td>
</tr>
<tr>
<td>Random Forest [12]</td>
<td>0.9009</td>
<td>0.9606</td>
<td>0.9740</td>
<td>0.9653</td>
<td>0.9768</td>
<td>0.9542</td>
</tr>
<tr>
<td>Real_Ada [23]</td>
<td>0.7995</td>
<td>0.8356</td>
<td>0.8037</td>
<td>0.7883</td>
<td>0.8368</td>
<td>0.8142</td>
</tr>
<tr>
<td>Dis_Ada [25]</td>
<td>0.7837</td>
<td>0.7547</td>
<td>0.8663</td>
<td>0.8535</td>
<td>0.9</td>
<td>0.8316</td>
</tr>
<tr>
<td>LogitBoost [40]</td>
<td>0.8879</td>
<td>0.9354</td>
<td>0.9532</td>
<td>0.9454</td>
<td>0.9701</td>
<td>0.9384</td>
</tr>
<tr>
<td>SVM [31]</td>
<td>0.8679</td>
<td>0.9291</td>
<td>0.9531</td>
<td>0.9244</td>
<td>0.9524</td>
<td>0.9249</td>
</tr>
<tr>
<td>Naive Bayes [86]</td>
<td>0.7652</td>
<td>0.8720</td>
<td>0.8768</td>
<td>0.8436</td>
<td>0.8504</td>
<td>0.8415</td>
</tr>
<tr>
<td>XGboost [14]</td>
<td>0.8995</td>
<td>0.9357</td>
<td>0.9606</td>
<td>0.9531</td>
<td>0.9759</td>
<td>0.9439</td>
</tr>
<tr>
<td>GBDT [10]</td>
<td>0.8621</td>
<td>0.9179</td>
<td>0.9580</td>
<td>0.9349</td>
<td>0.9770</td>
<td>0.9289</td>
</tr>
<tr>
<td>Ada_NB</td>
<td>0.8213</td>
<td>0.8673</td>
<td>0.9145</td>
<td>0.8960</td>
<td>0.9303</td>
<td>0.8864</td>
</tr>
<tr>
<td>Ada_SVM</td>
<td>0.7731</td>
<td>0.7486</td>
<td>0.8069</td>
<td>0.8001</td>
<td>0.8301</td>
<td>0.7919</td>
</tr>
<tr>
<td>Ada_PT</td>
<td>0.8886</td>
<td>0.9600</td>
<td>0.9699</td>
<td>0.9238</td>
<td>0.9611</td>
<td>0.9406</td>
</tr>
</tbody>
</table>

Table 3.6: Glass Dataset. The evaluation criterion is Area Under the curve of ROC (AUC) score.
3.5. Experimental Result

3.5.3 Depression Detection

In the CLPsych dataset, some data logs involve with sensitive information such as time zone, individual name, home address, has been filtered to protect the privacy of participants, the completeness of dataset might influence the analysis of depression online behaviours. I conduct main experiments of feature analysis on the TTDD dataset and verify the results on the CLPsych dataset.

In Tables 3.2 and 3.3, I summarise the classification results on the Twitter depression datasets. It is witnessed that the mean value, lower and upper limits of the IBPT’s AUC are higher than those of the other 12 methods and the IBPT achieves the highest AUC score with 0.93 and 0.83 separately on two datasets. In addition to analysing the effectiveness of the IBPT for depression classification, I further explore the feature importance in classification and compare the online behaviours between the depressed and normal users. I plot the feature importance of IBPT for classification on the TTDD, shown in Figure 3.6(a). The most crucial feature is LDA topics which seem to hold one-third of feature importance. I generate the statistics of the salient terms of LDA topics, shown in Figure 3.7(a), in order to discover the linguistic patterns of the Twitter users (LDA topics are multi-dimensional feature vectors which cannot be plotted using Kernel density estimation (KDE)). It should be noted that there are certain difference regarding the number of salient words between depressed and non-depressed classes because of the imbalance of the Twitter dataset (about 1:2 positive-negative ratio). The content of the tweets is so casual that both of the classes post tweets including

**Figure 3.4:** Feature importance list by IBPT for three toy datasets: (a) LSVT. (b) News. (c) Glasses.
Figure 3.5: Joint distributions against features using kernel density estimation (KDE), where contours represent the distribution of features and samples density become high with the contours color burning: (a) LSVT - more important features. (b) LSVT - less important features. (c) News - more important features. (d) News - less important features. (e) Glasses - more important features. (f) Glasses - less important features.
3.5. Experimental Result

many common used words such as ‘love’, ‘video’, ‘people’ in their tweets. Non-depressed persons are more likely to share their daily life on Twitter whose tweets include words such as ‘life’, ‘day’, ‘happi’. Different from the non-depressed users, the tweets’ topics of the depressed users include words ‘depress’, ‘mental health’, implying that the depressed users tend to post their mental status on Twitter.

Moreover, I wish to analyse the remaining features according to their importance. In Figure 3.8(a)-(b), the depressed and non-depressed users cannot be separated effectively as two classes mix up. This is due to the fact that except for the LDA topics, other features have small percentages in the feature map. I conduct the following experiments: (1) Figure 3.9(a) shows the difference of users’ posting proportions between the depressed and non-depressed classes at different time periods. I discover that the normal users tend to post more tweets in the daytime and the depressed users prefer to post tweets between 18 pm and 3 am, implying that the depressed users are likely to have insomnia. (2) Figure 3.9(b) shows the statistics of user posting patterns. The depressed users post 0.6 negative words and 1.4 positive words per tweet, which surpass those of the non-depressed users by 0.2 and 0.4, indicating that the depressed users may express their emotion and complain more about their bad moods. In addition, the non-depressed users post tweets including more emojis than the depressed users and prefer to mention other people in their posting. The depressed users’ tweets are more likely to be favoured by the others. (3) The features of user profile categories do not play a major role in classification. From Figure 3.9(c), I discover that the average counts of the followers, friends, and listed are similar between the depressed and non-depressed users, and the total favourites and the average count of the non-depressed class are larger than those of the depressed class. However, there is something in common between these features that the one class confidence interval is included in another class. As for the retweet count feature, the confidence interval of the non-depressed class does not fully include those of the depressed class. This is why retweet counts rank highly in the feature importance table. The user profile features may depend on the time length of the user spent on Twitter but they cannot effectively distinguish the non-depressed and depressed classes.

For comparison, I plot the Figure 3.6(b) to show the feature importance of IBPT on the CLPsych dataset and plot KDE Figure 3.10(a)-3.10(b) to show features joint distributions. Obviously, LDA topics still contribute most for classification.
Chapter 3. Inverse boosting pruning trees for depression detection on Twitter

Figure 3.6: Important feature list in the Twitter dataset. (a) The TTDD dataset. (b) The CLPsych dataset
3.5. Experimental Result

**Figure 3.7:** Salient words by LDA: (a) Normal class topic words. (b) Depression class topic words.

**Figure 3.8:** Twitter set - features joint distributions using kernel density estimation (KDE), where contours represent the distribution of features and samples density become high with the contours color burning: (a) Sample distribution against retweet counts and negative words. (b) Sample distribution against total favourites and emoticons.
Chapter 3. Inverse boosting pruning trees for depression detection on Twitter

![Figure 3.9](image)

**Figure 3.9:** Statistics of online behaviours of depression and non-depressed users. (a) Active time comparison. (b) Posting custom of each tweet. (c) Statistics of user profile features.

There is some difference of the remaining feature importance ranking between the TTDD and the CLPsych datasets. I summarize the statistical result of the CLPsych’s features in Table 3.7. In the LDA topics, ptsd and depressed users are more likely to post tweet related to ‘mental health’. Control users may share their daily activities on Twitter whose tweets including words such as ‘youtube’, ‘subscribe’. Different from the results of the TTDD, some features postive words, negative words or posting times show close average values between these three classes. As I have discussed above, it is caused by the discontinuity of data logs. I plot the curve, the radar and the bar figures of these features and show them in Figure 3.11(a)-3.11(c) to confirm my claim.

In order to verify the above predictions, I conduct experiments by combining different feature categories for classification and the result is presented in Figure 3.12(a)-(b). These figures show that linguistic features have the largest AUC area among the three feature categories and the user profile features’ AUC area is less than those of the other two categories. Compared with the single feature category, the combination of the multiple feature categories improves the classification performance. Finally, I compare my online depression detection framework with the present SOTA methods which I have listed in Section 2.4 (shown in Table 3.8). Methods used by Nadeem et al. and Song et al. only support binary classification so do not have the classification results on the CLPsych2015 dataset. My proposed method obtained the highest accuracy and F1 scores on the TTDD and the CLPsych2015 datasets separately, indicating that my proposed framework has good universality in different online depression detection datasets.
3.5. Experimental Result

**Figure 3.10**: The CLPsych dataset - features joint distributions using kernel density estimation (KDE), where contours represent the distribution of features and samples density become high with the contours color burning: (a) Sample distribution against retweet counts and negative words. (b) Sample distribution against total favourites and emoticons.

<table>
<thead>
<tr>
<th>LDA topics</th>
<th>Depression</th>
<th>PTSD</th>
<th>Control</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>nigga, harry, mentalhealth, health, depression, neuro, bruh, diagnose, loser, emotion</td>
<td>ebola, virus, ptsd, homeless, gaza, antidepressive, gaza, lmao, mentalhealth</td>
<td>artist, youtube, stats, review, ipad, blog, website, subscribe, daily, happy</td>
</tr>
<tr>
<td>favourites count</td>
<td>0.495</td>
<td>0.389</td>
<td>0.323</td>
</tr>
<tr>
<td>total favourites</td>
<td>3849</td>
<td>5014</td>
<td>1190</td>
</tr>
<tr>
<td>negative words</td>
<td>0.705</td>
<td>0.843</td>
<td>0.632</td>
</tr>
<tr>
<td>mention count</td>
<td>0.666</td>
<td>0.845</td>
<td>0.697</td>
</tr>
<tr>
<td>listed count</td>
<td>19.44</td>
<td>26.44</td>
<td>19.34</td>
</tr>
<tr>
<td>emoticons</td>
<td>3.434</td>
<td>0.841</td>
<td>3.373</td>
</tr>
<tr>
<td>positive words</td>
<td>1.29</td>
<td>1.48</td>
<td>1.21</td>
</tr>
<tr>
<td>post time count</td>
<td>0.357</td>
<td>0.325</td>
<td>0.338</td>
</tr>
<tr>
<td>friends count</td>
<td>1205</td>
<td>1158</td>
<td>654</td>
</tr>
<tr>
<td>post count</td>
<td>2202</td>
<td>2175</td>
<td>2051</td>
</tr>
<tr>
<td>retweet count</td>
<td>892</td>
<td>216</td>
<td>470</td>
</tr>
<tr>
<td>emojis</td>
<td>3.891</td>
<td>3.911</td>
<td>4.103</td>
</tr>
<tr>
<td>followers</td>
<td>2167</td>
<td>1538</td>
<td>1513</td>
</tr>
</tbody>
</table>

**Table 3.7**: Feature Statistics of the CLPsych dataset. The row of LDA topics present the top-10 most salient words in each class. The remain rows collect the average value of features in three classes.
Chapter 3. Inverse boosting pruning trees for depression detection on Twitter

**Figure 3.11:** Statistics of online behaviours of depression, ptsd and control users on the CLPsych dataset. (a) Active time comparison. (b) Posting custom of each tweet. (c) Statistics of user profile features.

**Figure 3.12:** ROC curves of different feature combinations. (a) The TTDD dataset. (b) The CLPsych dataset.
3.6. Discussion

In this chapter, I have made an attempt to automatically identify potential online depressed users. I presented an Inverse boosting pruning trees (IBPT) algorithm to classify non-depressed and depressed users. The IBPT outperformed several baselines against two depression databases. In the meantime, I verified the convergence of my algorithm IBPT through rigorous theoretical analysis with comprehensive experiments. Moreover, I utilised three UCI datasets to evaluate the classification ability of my method comprehensively, which showed my method outperforms the other baselines. I then analysed the feature importance of the IBPT and described the difference of the online behaviours of the non-depressed and the depressed classes. Finally, I used different combinations of the feature categories to confirm the effectiveness of the three feature categories for depression detection and my proposed framework outperformed several SOTA methods on depressed users classification.

But there are two weaknesses of the IBPT algorithm: (1) IBPT has high time complexity because of the pruning methods. More specifically, in each iteration, the base tree of the IBPT needs to grow fully with the training data then iteratively prune leaf nodes from bottom to top. This process improves the fitting and generalization ability of base trees but reduce its training speed. IBPT might not suit processing a large dataset. This problem may be further addressed by deploying the training of pruning trees on multiple CPUs or GPUs. (2) Because of the inverse boosting structure, the IBPT’s performance may be limited on the

<table>
<thead>
<tr>
<th>Methods\Datasets</th>
<th>TTDD</th>
<th>CLPsych2015</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>F1</td>
<td>acc</td>
</tr>
<tr>
<td>Park et al. [62]</td>
<td>0.69</td>
<td>0.71</td>
</tr>
<tr>
<td>Nadeem et al. [57]</td>
<td>0.70</td>
<td>0.73</td>
</tr>
<tr>
<td>Choudhury et al. [20]</td>
<td>0.78</td>
<td>0.79</td>
</tr>
<tr>
<td>Shuai et al. [74]</td>
<td>0.87</td>
<td>0.90</td>
</tr>
<tr>
<td>Jamil et al. [36]</td>
<td>0.77</td>
<td>0.74</td>
</tr>
<tr>
<td>Song et al. [75]</td>
<td>0.81</td>
<td>0.82</td>
</tr>
<tr>
<td>Shen et al. [73]</td>
<td>0.85</td>
<td>0.85</td>
</tr>
<tr>
<td>Cacheda et al. [13]</td>
<td>0.75</td>
<td>0.78</td>
</tr>
<tr>
<td>Ours</td>
<td><strong>0.89</strong></td>
<td><strong>0.90</strong></td>
</tr>
</tbody>
</table>

Table 3.8: Classification comparison with SOTA methods using F1 score and accuracy measures.
Chapter 3. Inverse boosting pruning trees for depression detection on Twitter

classification of some well labelled datasets. The boosting structure parameters should be adjusted according to specific datasets.

In the future, I expect to explore the semantic information of users’ posts by using the proposed algorithm and/or deep learning techniques and attempt to mine information from other source social networks, e.g. Facebook, Instagram, and Tumblr.
Chapter 4

Deep Neural Topic Model for Tweet Content Analysis

4.1 Introduction

In Chapter 2, I have introduced the previous researches using two NLP techniques (e.g. topic model, sentiment analysis) to extract features from social networks. Sentiment analysis is used to determine text polarity and topic modeling techniques are usually employed to extract document features and analyse posting content. Compared with other NLP techniques, an critical benefit of topic models is that they are unsupervised techniques which do not need human intervention in training process (i.e. sentiment analysis needs annotations of text polarity). Moreover, topic models can summarise the key information in the posting content of users.

Topic model is an unsupervised machine learning technique that is capable of scanning a set of documents, detecting word and phrase patterns, and automatically clustering word groups and similar expressions that best characterizes a set documents [7]. Intuitively, given a document related to a particular topic, I will expect some particular words to appear in the document: "apple" and "orange" will appear more often in documents for the fruit theme, "tiger" and "snake" will appear in documents for the animal theme, "is" and "are" have equal occurrences in documents. Generally, a document will include multiple topics with different proportions. A topic model captures this intuition in a mathematical framework to discover the distributions of topics in a document [80, 33, 7]. The document-generative process of topic models can be adapted to other kinds of analyses,
Chapter 4. Deep Neural Topic Model for Tweet Content Analysis

keeping only the analogy between document-topic-word and other kinds of objects [49].

In the online depression detection domain, the employed topic model approaches [21, 74, 73] are not the latest techniques which can be improved. Their established topic models cannot collect coherent document topics to describe depressed users online. On the other hand, from the experimental results of Chapter 3, I discovered that extracted features from the topic model LDA are crucial for depression detection. In this work, I focus on features development to improve the performance of depression detection. I wish to build a novel topic model which can produce interpretable document topics and representative document features.

In this chapter, I establish a novel topic model to mine meaningful text topics and distinguish document representations from tweets. The main contributions of this chapter have been listed in Section 1.3.

4.2 Review of recent topic models

Firstly, I review the workflow of Latent Dirichlet Allocation (LDA) because I employed LDA to extract document features in Chapter 3. LDA is an classical topic model which is shown in Figure 4.1. In Figure 4.1, a dirichlet distribution $\alpha$ represents a particular document and this topic distribution is $\theta$. A particular topic $Z$ can be selected from $\theta$, then the word distribution $\varphi$ of topic $Z$ will be randomly selected from the second dirichlet distribution $\beta$. $N$ is a generated word from $\varphi$ with topic $Z$. The following procedure is to use Gibbs sampling to maximize the loglikelihood of $p(W|Z, \varphi)$. In short, the LDA model is a clustering process which concentrates on word frequency and word distributions in corpus. Optimized $\theta$ can be represented as document vectors for classification tasks and word distribution $\varphi$ can be used to describe document content. Nevertheless, LDA has some shortages: (1) LDA employs Bag of Words (BoW) as the representations of the document input which just concentrates on word frequency but ignoring the sentence order and semantic information. (2) LDA employs Markov chain Monte Carlo (MCMC) or expectation maximization (EM) algorithm to approximate intractable distributions (e.g. documents and latent topics distribution) but these methods are difficult to converge and have high computational complexity.
4.2. Review of recent topic models

With the development of deep learning, many research works focus on combing deep neural network with topic models to address the computational complexity problem. Hinton et al. [32] proposed a probabilistic undirected topic model named Replicated Softmax Model (RSM) based on the Restricted Boltzmann machine [77] to model the distribution of Bag of Words (BoW) input. The framework of RSM is shown in Figure 4.2. The left figure denotes the inference process of RSM that BoW inputs $v_d$ are passing through a dense layer to infer latent topics $h_d$. Each input word $v_d$ share weight $w_d$ in the inference processing of latent topics. Then, as shown in the right figure of Figure 4.2, they employed latent topics to generate outputs matching with input words. The whole structure of RSM is regarded as a generative process. However, RSM directly models the distribution of latent high-dimensional vectors (i.e. latent topics) in a deep neural network is intractable. Deep Boltzmann machine [76] also employed BoW as a document input and used multiple hidden layers to encode inputs, where the model optimizes the parameters of intractable distributions by using stochastic gradient descent. Nevertheless, this method models the inference process of documents by employing multiple dense layers which reduces the generalisation ability of the model and yield poor performance on complicated and large corpus.

In order to learn good features from documents, Larochelle et al. [43] presented DocNADE to decompose the joint distributions of high-dimensional vectors into autoregressive conditional distributions. The framework of DocNADE is shown in Figure 4.3. Each document $D$ in the DocNADE is converted to a sequence $S = [v_1, v_2, v_3, ..., v_D]$ of size $D$, where each word index $v_d \in \{1, 2, 3, ..., V\}$ take a value from a fixed vocabulary $V$. Then DocNADE extracts latent topics.
$h_d$ from the word sequence $S$ through combining preceding observations $v_{<d} \in \{v_1, v_2, ..., v_{d-1}\}$ using a multilayer perceptron (MLP). Next, in the generative process, latent topics vector $h_d$ is used to predict the probability of each word in Vocabulary $V$ and compute the loss of cross entropy. The joint distribution of all words $p(v)$ is decomposed as a product of conditional distribution $p(v) = \prod_{d=1}^{D} p(v_d | v_{<d})$. Normally, the word sequence input for the DocNADE has removed stop words (e.g. "the", "a", "an", "in") to ensure the collected topics are interpretable. Word embedding process produces each word vector through analysing the word position and the co-occurrence relationship between its contextual words. So the initial word embeddings $W$ can not provide exact information of word meaning which leads DocNADE to performing poorly in processing complicated corpus.

Most recently, Gupta et al. [29] introduced a Bidirectional DocNADE model (iDocNADE) incorporated with Glove pre-trained word vectors [64] to address the data sparsity problems. Nevertheless, the proposed bidirectional structure model has a shortcoming in capturing syntactic features of sentences.
Figure 4.3: Document Neural Autoregressive Topic Model (DocNADE). Blue line represents hidden vectors $h_i$ share same weight in the generation of output. Image Courtesy of [44].
In this chapter, I propose two improved models based on DocNADE to improve the model performance on three aspects, generalisation ability, interpretability and applicability. I introduce the details of the proposed methods in the following section.

4.3 Proposed Method

In this section, I propose two improved models based on DocNADE: (1) FastText Distributional Prior DocNADE (fdp-DocNADE): I incorporate FastText embeddings with DocNADE to address complex corpus. (2) FastText Distributional Prior DocNADE with the Attention Mechanism (fdp-DocNADEa): I extract syntactic features via the Attention Mechanism based on fdp-DocNADE.

4.3.1 Proposed Model 1: FastText Distributional Prior DocNADE

My proposed model fdp-DocNADE is shown in Figure 4.4. Similar to DocNADE, fdp-DocNADE models each document $D$ as a word indices sequence $S$. Then I employ FastText pre-trained words embeddings \[37\] to introduce additional semantic information to my model, thereby enabling my model to produce more coherent topic distributions and better document representations.

Standard word vectors ignore the internal structure that contains rich information which is useful for computing representations of rare or misspelled words \[55\]. The user posting texts of my experimental Twitter datasets are random and usually contain some misspelled words, here I employ FastText word embeddings to supply pre-trained semantic information. Instead of learning word vectors directly, the FastText splits each word into n-grams characters to learn the internal structure information. For example, "intelligence" with n=3, the n-grams list of the FastText is $<$in, int, nte, tel, ell,lli,lig,ige,gen,enc,nce,ce>, where angular brackets denote the word beginning and ending. The output word vector of FastText is defined as follows:

$$E_{.,v_d} = v'_d + \frac{1}{|N|} \sum_{n \in N} x_n$$

(4.1)

where $v'_d$ is the standard word vector obtained from word2vec \[27\], $N$ is the pre-defined segmentation number of n-grams, $x_n$ is the embedding vector of n-grams.
4.3. Proposed Method

**Figure 4.4:** Schematic overview of fdp-DocNADE. The input and output (double circle) of each word $v_d$ are multinomial. Left green matrix $W$ represents the initial word embeddings and right red matrix $E$ is the introduced FastText pre-trained word embeddings. $w_d$ and $e_d$ are embedding vectors looking up from $W$ and $E$ respectively. $h_d \rightarrow$ denotes the forward network projected hidden units (latent topics) from the concat of $w_d$ and $e_d$. Blue dot lines with dots signify the shared weight of hidden units.
I concatenate word vectors from FastText embeddings $E_{:,v_d}$ with initial word vectors $W_{:,v_d}$ in a dense layer. The forward hidden units $h_d^\rightarrow(v_{<d})$ can be represented as:

$$h_d^\rightarrow(v_{<d}) = \begin{cases} \text{relu}(1^{T \times 1}), d = 1 \\ \text{relu}\left(\prod_{m=1}^{M} O_m \left(\sum_{i<d} \text{Proj}(W_{:,v_i} \oplus E_{:,v_i})\right) + b\right), d \neq 1 \end{cases}$$ (4.2)

where $W \in \mathbb{R}^{H \times V}$ denotes a word embedding matrix corresponding to the vocabulary $S$ and $W_{:,v_i}$ is a word vector from $W$, $O_m \in \mathbb{R}^{T \times H}$ is a hidden unit weight matrix, $M$ denotes the deep layers number of MLP and $b \in \mathbb{R}^T$ represents bias factor, $H$ is the embedding dimension of words, $T$ is the number of the pre-defined latent topics, relu is a Rectified Linear Unit activation function which has a good non-linear transformation ability, symbol $\oplus$ represents the concatenation operation between two word vectors, $\text{Proj}$ denotes a projection layer which projects $W_{:,v_i} \oplus E_{:,v_i}$ to dimension $\mathbb{R}^{H \times V}$, $h_1^\rightarrow \in 1^{T \times 1}$ is a bias matrix.

Next, in the generative process, latent topics vectors $h_d^\rightarrow$ are used to predict the probability of each word in the Vocabulary $V$ and compute the loss of cross entropy. The conditional distribution of each word $p(v_d | v_{<d})$ is defined as follows:

$$p(v_d | v_{<d}) = \frac{e^{h_n + G_n h_d^\rightarrow(v_{<d})}}{\sum_{n'} e^{h_n' + G_n h_d^\rightarrow(v_{<d})}}$$ (4.3)

where $G \in \mathbb{R}^{V \times T}$ is a fully-connected layer weight matrix which connects hidden topics with the output. Each document $D$ log-likelihood is given by $\log p(D) = \sum_{d=1}^{D} \log p(v_d | v_{<d})$. The proposed fdp-DocNADE algorithm scheme is illustrated in Algorithm 3.

### 4.3.2 Proposed Model 2: FastText Distributional Prior DocNADE with the Attention Mechanism

The original autoregressive structure of DocNADE is an accumulation process of word vectors from previous time-steps, and the produced hidden units $h$ can be viewed as a Fibonacci sequence. But this autoregressive structure can not capture effective syntactic features from sentences, accumulating word vectors directly
Algorithm 3 FastText Distributional Prior DocNADE.

**Input:** A training document $D$, Pre-trained FastText Embeddings $E$

**output:** Log-likelihood of all words $p(v)$

**Hyper-parameters:** Topic number $T$, embedding dimension $H$, deep layer number $M$

1. Convert document $D$ to a word indices sequence $S$ and look up each word in Word embedding matrix $W$.
2. Look up word in FastText Embeddings $E$.
3. Project $(W_{:,v_i} \oplus E_{:,v_i})$ to proper dimension $\mathbb{R}^{H \times V}$
4. for $d \in (1, D)$ do
   5. Infer latent topics $h_{d \rightarrow}$ using Eq. (4.2).
   6. Generate multinomial distribution and compute each output word conditional probability using Eq. (4.3).
7. end for
8. $p(v) \leftarrow \prod_{d=1}^{D} p(v_d | v_{<d})$.

Attention Mechanism, motivated by how I pay attention to correlate words in one sentence. For example, as shown in Figure 4.6, when seeing "eating", I expect to encounter a food word rather than a color word [82]. Word "eating" should come with different weights against the prediction of "green" and "apple" separately. Attention in the deep neural network can be broadly interpreted as an importance weighting vector.

Examining fdp-DocNADE, I concatenate FastText vectors $E_{:,v_d}$ with initial word vectors $W_{:,v_d}$ to compute the hidden vector $h_{d \rightarrow}$. Here, I introduce an attention layer between the hidden layer and the output to reassign prediction weights for the hidden vector $h_{d \rightarrow}$. In the attention layer, each vector is assigned a new global weight according to its position in the sentence and the contextual relationship with preceding vectors:

$$a_{d,i} = \frac{e^{\text{score}(h_{d \rightarrow}, h_{i \rightarrow})}}{\sum_{i=1}^{d} e^{\text{score}(h_{d \rightarrow}, h_{i \rightarrow})}}$$  (4.4)
Chapter 4. Deep Neural Topic Model for Tweet Content Analysis

Figure 4.5: Schematic overview of fdp-DocNADEa. I add an attention layer between hidden layer and output to reassign the weight of each unit $h^d_j$ based on syntactic information. Grey line $a_d$ is the reassigned weight for each hidden units $h^d_j$. $C_d$ denotes a context vector which includes context information of $h_d$ hidden unit.

Figure 4.6: One word “attends” to other words in the same sentence differently. Image courtesy of [82].
where $a_{d,i}$ represents the global aligned weight of the hidden vector $h_i$ corresponding to $d$th output. The alignment score $\text{score}(h_d^\rightarrow, h_i^\rightarrow)$ is defined as follows:

$$\text{score}(h_d^\rightarrow, h_i^\rightarrow) = \sigma(U_a \cdot \tanh(U_d h_d^\rightarrow + U_i h_i^\rightarrow) + b_a)$$  \hspace{1cm} (4.5)$$

where $\sigma$ is a self-defined activation function, $U_a, U_d$ and $U_i$ are weight matrices of dense layers, $\tanh$ denotes a tanh activation function and $b_a$ is a bias factor. The context vector $C_d$ for the $d$th output is the combination of the re-weighted preceding hidden vectors:

$$C_d = \sum_{i=1}^{d} a_{d,i} h_i^\rightarrow$$  \hspace{1cm} (4.6)$$

where the hidden vector $h_d^\rightarrow$ includes word semantic information and the context vector $C_d$ has collected syntactic features, and I sum these two vectors to produce the conditional distribution of each word $p(v_d | v_{<d})$, thereby Eq. (4.3) can be transformed as follows:

$$p(v_d | v_{<d}) = \frac{\exp(b_n + G_{n,:} \cdot (h_d^\rightarrow(v_{<d}) + C_d))}{\sum_{n'} \exp(b_{n'} + G_{n',:} \cdot (h_d^\rightarrow(v_{<d}) + C_d))}$$  \hspace{1cm} (4.7)$$

where $G$ represents the distribution between the latent topics and the vocabulary, it is an important evaluation point for the topic coherent of my topic model. The final state of $h_D^\rightarrow(v_{<D}) + C_D$ is the trained document $D$ representation which can be used for text classification directly. The algorithm scheme of fdp-DocNADEa is illustrated in Algorithm 4.

### 4.4 Experiment Setup

#### 4.4.1 Datasets

In my experimental work, I use five text databases to evaluate my proposed model’s performance:

(1) **TMN**: TagMyNews dataset is a collection of datasets of short text fragments [65]. This is a dataset of 32K english news extracted from RSS feeds of popular newspaper websites (nyt.com, usatoday.com, reuters.com). TMN dataset includes 9 categories: Sport, Business, U.S., Health, Sci and Tech, World and Entertainment.
Chapter 4. Deep Neural Topic Model for Tweet Content Analysis

Algorithm 4 FastText Distributional Prior DocNADE with Attention Mechanism.

Input: A training document \( D \), Pre-trained FastText Embeddings \( E \)

output: Log-likelihood of all words \( p(v) \)

Hyper-parameters: Topic number \( T \), embedding dimension \( H \), deep layer number \( M \)

1: Convert document \( D \) to a word indices sequence \( S \) and look up each word in Word embedding matrix \( W \).
2: Look up word in FastText Embeddings \( E \).
3: Project \( (W_{:,\nu} \oplus E_{:,\nu}) \) to proper dimension \( \mathbb{R}^{H \times V} \)
4: for \( d \in (1, D) \) do
5: Infer latent topics \( h_{d}\rightarrow \) using Eq. (4.2).
6: Compute the alignment score between \( h_{d}\rightarrow \) and \( h_{i}\rightarrow \) by Eq. (4.5)
7: Reassign the global align weight of hidden vector \( h_{i} \) corresponding to \( d \)th output using Eq. (4.4)
8: Combine re-weighted preceding hidden vectors to produce context vectors \( C_{d} \) by Eq. (4.6)
9: Combine hidden vectors \( h_{d}\rightarrow \) with context vectors \( C_{d} \) to compute each output word conditional probability using Eq. (4.7).
10: end for
11: \( p(v) \leftarrow \prod_{d=1}^{D} p(v_{d} | v_{<d}) \).

(2) TREC6: TREC6 dataset harvested a set of English test questions and split questions into 6 types that whether the question is about person, location, numeric information [47].

(3) 20NS: The 20 Newsgroup dataset is a collection of approximately 20,000 newsgroup documents [42]. This dataset includes 20 different classes, each corresponding to a different topic.

(4) TTDD: The Tsinghua Twitter depression detection dataset I have gave the details clearly in the Chapter 3.

(5) CLPsych: The CLPsych 2015 Twitter depression detection dataset I have gave the details clearly in the Chapter 3.

Statistics of five datasets are shown in Table 4.1.

4.4.2 baselines comparison methods

DocNADE: I have introduced DocNADE above. I compare my proposed models with the baseline DocNADE to verify the working of my improvement on DocNADE.
4.4. Experiment Setup

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Train</th>
<th>Validation</th>
<th>Test</th>
<th>V</th>
<th>L</th>
<th>C</th>
<th>Domain</th>
</tr>
</thead>
<tbody>
<tr>
<td>TMN</td>
<td>22.8k</td>
<td>2.0k</td>
<td>7.8k</td>
<td>2k</td>
<td>19</td>
<td>7</td>
<td>News</td>
</tr>
<tr>
<td>TREC6</td>
<td>5.5k</td>
<td>0.5k</td>
<td>0.5k</td>
<td>2k</td>
<td>10</td>
<td>6</td>
<td>Q&amp;A</td>
</tr>
<tr>
<td>20NS</td>
<td>8.9k</td>
<td>2.2k</td>
<td>7.4k</td>
<td>2k</td>
<td>229</td>
<td>20</td>
<td>News</td>
</tr>
<tr>
<td>TTDD</td>
<td>4.5k</td>
<td>1.5k</td>
<td>1.5k</td>
<td>15k</td>
<td>318</td>
<td>2</td>
<td>Tweets</td>
</tr>
<tr>
<td>CLPsych</td>
<td>1.1k</td>
<td>0.3k</td>
<td>0.3k</td>
<td>40k</td>
<td>1200</td>
<td>3</td>
<td>Tweets</td>
</tr>
</tbody>
</table>

Table 4.1: Data Statistics. The column of Train Validation and Test represent the training, testing and validation set size of each database. The symbol V: Vocabulary Size, L: Average Text length, C: Number of Classes. Domain means the data source of each database.

iDocNADEe: Document Informed Neural Autoregressive Distribution Estimator with Embeddings (iDocNADEe) [29] combines a bidirectional autoregressive structure with the origin DocNADE to capture forward and backward syntactic information of sentences.


NVDM: Neural Variational Document model [54] introduced a generic variational inference framework to learn continuous stochastic document representations from BoW.

LDA: Latent Dirichlet allocation is a typical traditional topic modelling algorithm proposed in [7].

HDP: The hierarchical Dirichlet process (HDP) is a Bayesian nonparametric model that can be used to model mixed-membership data with a potentially infinite number of components [81]. It has been applied widely in probabilistic topic modeling.

I compare my proposed models (fdp-DocNADE, fdp-DocNADEa) with the above methods for three aspects (Generalization, Interpretability, Applicability) in the following section.
Chapter 4. Deep Neural Topic Model for Tweet Content Analysis

4.5 Experimental Result

In this section, I evaluate my proposed topic models (fdp-DocNADE, fdp-DocNADEa) firstly by computing the models’ perplexity. Then I assess the interpretation ability of the captured topics by computing the topic coherence. I conduct experiments to evaluate the applicability of the proposed models. Finally, I validate the classification ability of the presented models against the target "Twitter depressed users detection”.

4.5.1 Model Generalization Evaluation

Perplexity is a criterion of how well a probability distribution or probability model predicts a sample [44, 32, 29]. The formulation of perplexity $PPL$ [7] is shown as follows:

$$PPL = 2^{-\frac{1}{N} \sum_{n=1}^{N} \frac{1}{|v^n|} \log p(v^n)}$$

(4.8)

where $N$ is the total number of documents and $|v^n|$ denotes the word number in document $v^n$, the log-likelihood $\log p(v^n)$ represents the joint distribution of document $v^n$ which equals to each model’s prediction. So a good topic model is expected to have a high log-likelihood of document predictions with a low perplexity value.

<table>
<thead>
<tr>
<th>BaselineDatasets</th>
<th>TMN</th>
<th>TREC6</th>
<th>20NS</th>
<th>TTDD</th>
<th>CLPsych</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Traditional Topic Models</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LDA</td>
<td>8184</td>
<td>6502</td>
<td>2208</td>
<td>7987</td>
<td>14898</td>
</tr>
<tr>
<td>HDP</td>
<td>3135</td>
<td>2164</td>
<td>3157</td>
<td>5437</td>
<td>14263</td>
</tr>
<tr>
<td><strong>Deep Neural Topic Models</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NVDM</td>
<td>711</td>
<td>117</td>
<td>836</td>
<td>4590</td>
<td>13057</td>
</tr>
<tr>
<td>DocNADE</td>
<td>592</td>
<td>64</td>
<td>894</td>
<td>4678</td>
<td>12986</td>
</tr>
<tr>
<td>ctx-DeepDNEe</td>
<td>1427</td>
<td>276</td>
<td>648</td>
<td>4117</td>
<td>9978</td>
</tr>
<tr>
<td>iDocNADEe</td>
<td>640</td>
<td>61</td>
<td>731</td>
<td>4359</td>
<td>10964</td>
</tr>
<tr>
<td>fdp-DocNADE</td>
<td>563</td>
<td>62</td>
<td>748</td>
<td>4295</td>
<td>11898</td>
</tr>
<tr>
<td>fdp-DocNADEa</td>
<td>402</td>
<td>91</td>
<td>588</td>
<td>3796</td>
<td>9743</td>
</tr>
</tbody>
</table>

TABLE 4.2: Statistics of perplexity value. LDA and HDP belong to traditional machine learning topic model. The remaining methods are included in deep learning based topic models.

The computed perplexity results on four datasets are presented in Table 4.2. It is witnessed that my proposed method fdp-DocNADE achieves lower perplexity than the original DocNADE over the four datasets, which confirms the external
knowledge FastText embeddings improve the generalization ability of the original model. My second improved model fdp-DocNADEa outperforms the other methods on the four datasets, and the perplexity result of fdp-DocNADEa on the TREC6 dataset is limited because the average text length of the TREC6 is too short which can not provide sufficient syntactic information. It is evident that the deep neural topic models achieve better results than the traditional methods (LDA, HDP) in the generalization evaluation.

### 4.5.2 Interpretability of Captured Topics

Topic coherence is a criterion which is used to evaluate the interpretability of topic models [54, 69]. My topic model has a latent topic matrix \( G \in \mathbb{R}^{T \times V} \), which represents the word distribution on the captured topics. Each captured topic consists of words, and the topic coherence [59] is applied to the top \( N \) words from the topic. It is defined as follows:

\[
CoherenceScore = \sum_{i<j} \log \frac{p(v_i, v_j)}{p(v_i)p(v_j)}
\]  

(4.9)

where,

\[
p(v_i) = \frac{e_{doc}(v_i)}{e_{doc}} \quad \frac{e_{doc}(v_i, v_j)}{e_{doc}}
\]  

(4.10)

\( e_{doc} \) is the number of documents, \( e_{doc}(v_i) \) means the number of documents where word \( v_i \) appears. The numerator \( e_{doc}(v_i, v_j) \) is the number of documents where words \( v_i \) and \( v_j \) appear together. The topic coherence can be regarded as the average of the pairwise word-similarity scores of the words in the topic. Good topics with high coherence scores can be described by a short label. To explain the topic coherence intuitively, I list three models’ top score topics from 20NS dataset in Table 4.3. Obviously, the topic from my proposed model fdp-DocNADEa is related to ‘religion’ theme and all the words’ meanings seem coherent. The topic words from DocNADE and LDA are less coherent than fdp-DocNADEa’s results.

Here I use a gensim module to evaluate the coherence of 200 topics collected by the topic models, and the computed scores are shown in Table 4.4. I have obtained the following observations: (1) My proposed model fdp-DocNADEa shows competitive results compared with SOTA methods (e.g. ctx-DeepDNEe,
Chapter 4. Deep Neural Topic Model for Tweet Content Analysis

<table>
<thead>
<tr>
<th>fdp-DocNADEa</th>
<th>DocNADE</th>
<th>LDA</th>
</tr>
</thead>
<tbody>
<tr>
<td>church, atheists, christian, belief, pray, atheism, buddhism, bible, temple, catholicism</td>
<td>scripture, church, christian, lord, jesus, follow, christ, cross, table, bible</td>
<td>debris, johnson, center, larc, space, gov, nasa, follow, guy, question</td>
</tr>
<tr>
<td>0.846</td>
<td>0.814</td>
<td>0.769</td>
</tr>
</tbody>
</table>

Table 4.3: Top coherence score topics with top 10 words in 20NS

<table>
<thead>
<tr>
<th>Baseline\Datasets</th>
<th>TMN</th>
<th>TREC6</th>
<th>20NS</th>
<th>TTDD</th>
<th>CLPsych</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>W10</td>
<td>W20</td>
<td>W10</td>
<td>W20</td>
<td>W10</td>
</tr>
<tr>
<td>LDA</td>
<td>0.461</td>
<td>0.509</td>
<td>0.333</td>
<td>0.426</td>
<td>0.602</td>
</tr>
<tr>
<td>HDP</td>
<td>0.661</td>
<td>0.750</td>
<td>0.644</td>
<td>0.789</td>
<td>0.741</td>
</tr>
<tr>
<td>NVDM</td>
<td>0.712</td>
<td>0.745</td>
<td>0.701</td>
<td>0.837</td>
<td>0.757</td>
</tr>
<tr>
<td>DocNADE</td>
<td>0.709</td>
<td>0.811</td>
<td>0.746</td>
<td>0.860</td>
<td>0.713</td>
</tr>
<tr>
<td>ctx-DeepDNEe</td>
<td>0.709</td>
<td>0.825</td>
<td>0.713</td>
<td>0.809</td>
<td>0.696</td>
</tr>
<tr>
<td>iDocNADEe</td>
<td>0.771</td>
<td>0.805</td>
<td>0.752</td>
<td>0.866</td>
<td>0.748</td>
</tr>
<tr>
<td>fdp-DocNADE</td>
<td>0.717</td>
<td>0.818</td>
<td>0.753</td>
<td>0.858</td>
<td>0.729</td>
</tr>
<tr>
<td>fdp-DocNADEb</td>
<td>0.711</td>
<td>0.801</td>
<td>0.748</td>
<td>0.837</td>
<td>0.747</td>
</tr>
</tbody>
</table>

Table 4.4: Statistics of average topic coherence with top ten (W10) and twenty (W20) words.

iDocNADEe), fdp-DocNADEa achieves good coherence score on the three long-text datasets (20NS, TTDD, CLPsych) which verifies the benefits of attention layer on large scale corpus. (2) fdp-DocNADE outperforms the DocNADE on all the datasets, which confirms the introduced FastText embeddings improves the model’s clustering ability. (3) My proposed model fdp-DocNADEa does not perform well over the coherence evaluation on the TMN and the TREC6 datasets. The results are due to the fact that the attention mechanism cannot capture sufficient syntactic features from the short texts.

4.5.3 Applicability Testing

Furthermore, a good topic model is also expected to learn good document representations which can be used for downstream tasks, like text classification. In order to evaluate the quality of the learned document representations from the topic models, I conduct text classification tasks against the topic models on five datasets. I use the experimental setup similar to Gupta [29], where the dimension of the document representations is 200. I use these vectors to train a logistic regression classifier [63] and evaluate their classification performance. I measure
the F1 score and the accuracy of the text classification which is presented in Table 4.5. My proposed models show competitive classification performance on the five datasets. It is witnessed that the introduced FastText embeddings improves the classification performance of the original DocNADE. Although fdp-DocNADEa does not perform well on small scale datasets (e.g. TMN and TREC6) because of the introduced attention mechanism. fdp-DocNADEa outperforms fdp-DocNADE on the 20NS, TTDD and CLPsych datasets which verifies the effectiveness of the attention mechanism on large corpus.

<table>
<thead>
<tr>
<th>Baseline\Datasets</th>
<th>TMN F1 acc</th>
<th>TREC6 F1 acc</th>
<th>20NS F1 acc</th>
<th>TTDD F1 acc</th>
<th>CLPsych F1 acc</th>
</tr>
</thead>
<tbody>
<tr>
<td>LDA</td>
<td>0.571</td>
<td>0.642</td>
<td>0.517</td>
<td>0.618</td>
<td>0.143</td>
</tr>
<tr>
<td>HDP</td>
<td>0.190</td>
<td>0.314</td>
<td>0.313</td>
<td>0.411</td>
<td>0.086</td>
</tr>
<tr>
<td>NVDM</td>
<td>0.712</td>
<td>0.745</td>
<td>0.824</td>
<td>0.837</td>
<td>0.514</td>
</tr>
<tr>
<td>DocNADE</td>
<td>0.740</td>
<td>0.778</td>
<td>0.804</td>
<td>0.822</td>
<td>0.428</td>
</tr>
<tr>
<td>ctx-DeepDNEe</td>
<td>0.707</td>
<td>0.726</td>
<td>0.670</td>
<td>0.694</td>
<td>0.524</td>
</tr>
<tr>
<td>iDocNADEe</td>
<td>0.771</td>
<td>0.805</td>
<td>0.842</td>
<td>0.844</td>
<td>0.518</td>
</tr>
<tr>
<td>fdp-DocNADE</td>
<td>0.778</td>
<td>0.804</td>
<td>0.854</td>
<td>0.858</td>
<td>0.490</td>
</tr>
<tr>
<td>fdp-DocNADEa</td>
<td>0.749</td>
<td>0.790</td>
<td>0.805</td>
<td>0.826</td>
<td>0.520</td>
</tr>
</tbody>
</table>

Table 4.5: Text classification with F1 score and accuracy measures.

I plot curves of the f1 scores and the accuracy changing with the training data percentages over the TTDD and the CLPsych datasets, which are shown in Figure 4.7(a)- 4.7(b) and Figure 4.8(a)- 4.8(b). I observed that the performance of the LDA, the NVDM and the ctx-DeepDNE models have large variations with the increasing of the training data percentages. My proposed model fdp-DocNADE outperforms the baseline DocNADE which proves the introduced FastText embeddings prior is beneficial for improving the original model’s classification ability. fdp-DocNADEa achieves the best classification performance when trained with all the training data.

4.5.4 Twitter depressed users detection

In the above experiments, my proposed models achieve competitive results compared with the other SOTA methods, especially the fdp-DocNADEa obtains good performance over the topic coherence and the classification evaluation on the two Twitter datasets. Here, I employ the fdp-DocNADEa to analyse the tweet content of the depressed users.
Figure 4.7: Evaluating the F1 score and the accuracy with the different training data percentages of the TTDD dataset (a) The F1 score curve (b) The accuracy curve.
4.5. Experimental Result

Figure 4.8: Evaluating the F1 score and the accuracy with the different training data percentages of the CLPsych dataset (a) The F1 score curve (b) The accuracy curve.
Figure 4.9: Document representations distribution using TSNE on TTDD dataset. (a): Document vectors learned from LDA. (b): Document vectors learned from DocNADE. (c): Document vectors learned from fdp-DocNADE. (d): Document vectors learned from fdp-DocNADEa.
4.5. Experimental Result

Firstly, I use t-Distributed Stochastic Neighbor Embedding (TSNE) to reduce the dimensions of the document vectors and plot the vectors’ distribution over the TTDD dataset in Figure 4.9 compared with the results of the LDA, the Doc-NADE and the fdp-DocNADE. In Figure 4.9, because the dimensions of the extracted document vectors are 200, projecting the data onto 3D space cannot distinguish two classes clearly. It still can be discovered that the learned document vectors of LDA cannot separate the non-depressed and the depressed groups well and the two classes’ points are overlapped each other. The fdp-DocNADEa produced vectors show that these two classes can be separated, which proves the superiority of my proposed method. The similar results also can be found on the CLPsych dataset which is shown in Figure 4.10. To analyse the collected text features comprehensively, I compute the average difference of topic weight values between the non-depressed and the depressed classes.

Figure 4.11(a) shows top four topics with the highest average difference absolute weight between non-depressed and depressed class. (b)CLPsych set: Top four topics with highest average difference absolute weight between control, depression and ptsd users

Figure 4.11(a) shows top four topics with the highest average difference of weight values over the TTDD dataset. I discover that topics 1 and 3 are good indicators of the depressed class. Topics 2 and 4 are more related to the non-depressed users’ daily posts. In Figure 4.11(b), the PTSD class holds the highest weight in topic 1 and depressed users are more likely to post tweets related to topics 2 and 3. I use wordcloud to show top probability words from four topics in Figure 4.12 over the TTDD dataset. Topic 1 includes some words such as ‘mental health’, ‘illness’, ‘bipolar ’ which are related to the ‘Mental health’ theme. Topic 3
4.5. Experimental Result

Figure 4.12: Top probability words in four topics on the TTDD set with manual defined 'theme'

...collects 'stream', 'youtube', 'minecraft' which implies depressed users may talk about games and videos on Twitter. Topics 2 and 4 are related to the 'Politics' and 'idols' themes separately which implies normal users may pay attention to hot news. Figure 4.13 demonstrates the hot topics over the CLPsych dataset. In the topics 1 and 2, I find that mental illness sufferers (i.e. PTSD and depressed people) are more likely to share their mental conditions on Twitter.

Finally, I employ the extracted features from the topic models to train my proposed classifier IBPT and the results are shown in Figure 4.14(a)- 4.14(b). I combine the fdp-DocNADEa with the IBPT to form an entire framework which achieves the best classification performance with the auc scores 0.89 and 0.73 over the two Twitter datasets respectively, which outperforms the Chapter 3 results (e.g. using the LDA to extract document vectors). I update the results of the
Figure 4.13: Top probability words in four topics on the CLPsych set with manual defined ‘theme’
compared SOTA methods in Table 4.6. The results manifest that my proposed framework is promising for identifying Twitter users with depression.

Figure 4.14: Roc curve of the Inverse Boosting Pruning Trees training with topic models document representations (a) Roc of the TTDD set (b) Roc of the CCLPsych set

4.6 Discussion

In this chapter, I proposed a novel topic model to extract text features and interpretable topics from Twitter texts. I combined the FastText word embeddings with the baseline DocNADE to introduce external semantic knowledge. Then I used an Attention layer to reassign the hidden vectors’ weights for the predictions, and the proposed model was named as fdp-DocNADEa. I conducted experiments to evaluate the performance of the proposed method over three aspects on five publicly accessible databases. fdp-DocNADEa achieved an outstanding performance on the perplexity evaluation which proves the introduced attention mechanism is beneficial for improving model’s generalization ability. On the evaluation of the topic coherence, fdp-DocNADEa obtained competitive results compared with the present SOTA methods (iDocNADE, ctx-DeepDNEe). fdp-DocNADEa showed a strong clustering ability on the large corpus. In the models’ applicability testing, the text features produced by fdp-DocNADEa can help LR
model to achieve good classification results. Moreover, I used fdp-DocNADEa to collect document topics over the two Twitter datasets and discovered that depressed users are more likely to share their medical conditions on Twitter. Finally, I combined the fdp-DocNADEa with my proposed classifier IBPT to form an entire framework for Twitter depressed users classification and the results outperformed the present SOTA frameworks on the two Twitter datasets.

However, fdp-DocNADEa has shortcomings on processing short texts. In order to collect coherent topics, I filter the stop words in the experimental datasets. In the large scale datasets (e.g. 20NS, TTDD, CLPsych), long texts can support the attention layer to capture accurate syntactic features. But the attention mechanism cannot achieve consistent performance on the short texts datasets because there is no sufficient word co-occurrences for analysing. Inspired by [30, 22], the problem may be addressed through introducing a LSTM network to enhance the contextual relationships between sentences and establishing an extra network branch to compute the loss of the stop words.

<table>
<thead>
<tr>
<th>Methods\Datasets</th>
<th>TTDD F1</th>
<th>acc</th>
<th>CLPsych2015 F1</th>
<th>acc</th>
</tr>
</thead>
<tbody>
<tr>
<td>Park et al. [62]</td>
<td>0.69</td>
<td>0.71</td>
<td>0.61</td>
<td>0.57</td>
</tr>
<tr>
<td>Nadeem et al. [57]</td>
<td>0.70</td>
<td>0.73</td>
<td>~</td>
<td>~</td>
</tr>
<tr>
<td>Choudhury et al. [20]</td>
<td>0.78</td>
<td>0.79</td>
<td>0.77</td>
<td>0.76</td>
</tr>
<tr>
<td>Shuai et al. [74]</td>
<td>0.87</td>
<td>0.90</td>
<td>0.76</td>
<td>0.78</td>
</tr>
<tr>
<td>Jamil et al. [36]</td>
<td>0.77</td>
<td>0.74</td>
<td>0.66</td>
<td>0.61</td>
</tr>
<tr>
<td>Song et al. [75]</td>
<td>0.81</td>
<td>0.82</td>
<td>~</td>
<td>~</td>
</tr>
<tr>
<td>Shen et al. [73]</td>
<td>0.85</td>
<td>0.85</td>
<td>0.73</td>
<td>0.75</td>
</tr>
<tr>
<td>Cacheda et al. [13]</td>
<td>0.75</td>
<td>0.78</td>
<td>0.71</td>
<td>0.72</td>
</tr>
<tr>
<td>Ours(Chapter 3 method)</td>
<td>0.89</td>
<td>0.90</td>
<td>0.77</td>
<td>0.81</td>
</tr>
<tr>
<td>Ours(Chapter 4 method)</td>
<td><strong>0.91</strong></td>
<td><strong>0.91</strong></td>
<td><strong>0.80</strong></td>
<td><strong>0.82</strong></td>
</tr>
</tbody>
</table>

TABLE 4.6: Classification comparison with SOTA methods using F1 scores and accuracy measures.
Chapter 5

Conclusion and Future Works

5.1 Conclusion

In this thesis, my main research goal is to establish an automatic online depression detection system which can determine whether or not a social network user has exhibited signs of depression. I propose several novel methods on feature extraction and classification to improve the performance of detecting Twitter depressed users. I conclude my contributions below:

- **Inverse boosting pruning trees for depression detection on Twitter**: I have made an attempt to automatically identify potential online depressed users. I presented an Inverse boosting pruning trees (IBPT) algorithm to classify non-depressed and depressed users on Twitter. The IBPT outperformed several baselines against two depression databases. In the meantime, I verified the convergence of my algorithm IBPT through rigorous theoretical analysis with comprehensive experiments. Moreover, I utilised three UCI datasets to evaluate the classification ability of my method comprehensively, which showed my method outperforms the other baselines. I then analysed the feature importance of the IBPT and described the difference of the online behaviours of the non-depressed and the depressed classes. Finally, I used different combinations of the feature categories to confirm the effectiveness of the three feature categories for depression detection and my proposed framework outperformed several SOTA methods on depressed users classification.

- **Deep Neural Topic Model for Tweet Content Analysis**: I proposed a novel topic model to extract text features and interpretable topics from Twitter texts. I introduce the FastText word embeddings to the baseline DocNADE
to supply the semantic knowledge. Then I used a Attention layer to re-assign the hidden vectors’ weights for the predictions, and the proposed model was named as fdp-DocNADEa. I conducted experiments to evaluate the performance of the proposed method over three aspects on five publicly accessible databases. fdp-DocNADEa achieved outstanding performance on the perplexity evaluation which proves the introduced attention mechanism is beneficial to improving model’s generalization ability. On the evaluation of the topic coherence, the fdp-DocNADEa obtained competitive results compared with the present SOTA methods (iDocNADE, ctx-DeepDNEe). fdp-DocNADEa showed a strong clustering ability on the large corpus. In the models’ applicability testing, the text features produced by fdp-DocNADEa lead LR in the classification performance. Moreover, I used fdp-DocNADEa to collect document topics over the two Twitter datasets and discovered that depressed users are more likely to share their medical conditions on Twitter. Finally, I combined the fdp-DocNADEa with my proposed classifier IBPT to form an entire framework for Twitter depressed users classification and the results manifested that my proposed framework is promising for identifying Twitter users with depression.

5.2 Future Works

Several future research avenues appear at the end of this thesis:

(1) **Attempting to mine information from other social networks:** At present, I only find two publicly accessible Twitter datasets. I have not verified the effectiveness of our proposed method on processing other social networks’ data. In future, I may collect data logs from other social networks such as Facebook, Tumblr to establish a well-labelled dataset for analysing. It is critical to verify that the universality of my proposed method can be applied on different data.

(2) **Feature development on social networks’ posting images and emojis:** In this study, I removed emojis or pictures when I analysed tweet texts. In some way, emojis or pictures also contain some information about users’ emotion. I will explore these information using computer vision and natural language processing techniques. For instance, people may share their selfies on social
networks, I may utilize computer vision techniques to recognize users’ face expressions and static scenes, then link these with textual features to analyse users’ behaviours.

(3) **Cooperating with psychological industries for comprehensive research**: Because of individuals privacy protection, I cannot be certain whether or not these individuals suffer from depression. Further research needs to cooperate with local psychological industries to gain more comprehensive information of depression sufferers then establish the relationships between their social networks messages and severity of depression.

With the development of science and technology, I believe that an automatic online depression detection system has great potential to help timely diagnose and treat depression.
Appendix A

Ethics Approval Letter

University Ethics Sub-Committee for Science and Engineering and Arts Humanities
13/11/2019
Ethics Reference: 19816-lt228-se/in:informatics
TO: Name of Researcher Applicant: Lei Tong Department: Computer Science
Research Project Title: Depression Detection Via Twitter
Dear Lei Tong,
RE: Ethics review of Research Study application
The University Ethics Sub-Committee for Science and Engineering and Arts Humanities has reviewed and discussed the above application.

1. Ethical opinion
The Sub-Committee grants ethical approval to the above research project on the basis described in the application form and supporting documentation, subject to the conditions specified below.

2. Summary of ethics review discussion
The Committee noted the following issues: This application has been approved on condition that the further anonymisation procedures outlined in the application are implemented on the dataset.

Best wishes, Martin Phillips

3. General conditions of the ethical approval
The ethics approval is subject to the following general conditions being met prior to the start of the project:
As the Principal Investigator, you are expected to deliver the research project in accordance with the University’s policies and procedures, which includes the University’s Research Code of Conduct and the University’s Research Ethics Policy.
Appendix A. Ethics Approval Letter

If relevant, management permission or approval (gate keeper role) must be obtained from host organisation prior to the start of the study at the site concerned.

4. Reporting requirements after ethical approval

You are expected to notify the Sub-Committee about:
• Significant amendments to the project
• Serious breaches of the protocol
• Annual progress reports
• Notifying the end of the study

5. Use of application information

Details from your ethics application will be stored on the University Ethics Online System. With your permission, the Sub-Committee may wish to use parts of the application in an anonymised format for training or sharing best practice. Please let me know if you do not want the application details to be used in this manner.

Best wishes for the success of this research project.

Yours sincerely,

Dr. Martin Phillips Chair
Bibliography


[23] Paul K Edwards, Dina Duhon, and Suhail Shergill. “Real AdaBoost: boosting for credit scorecards and similarity to WOE logistic regression”. In: ().


[74] Hong-Han Shuai et al. “A Comprehensive Study on Social Network Mental Disorders Detection via Online Social Media Mining”. In: IEEE Transactions on Knowledge and Data Engineering 30.7 (2018), pp. 1212–1225.


