LEARNING AND INFERRING USER CHARACTERISTICS FROM ONLINE BEHAVIOR AND CONTENT

A Dissertation

Submitted to the Graduate School
of the University of Notre Dame
in Partial Fulfillment of the Requirements
for the Degree of

Doctor of Philosophy

by

Munira Syed

Nitesh V. Chawla , Director

Graduate Program in Computer Science and Engineering
Notre Dame, Indiana

July 2020

LEARNING AND INFERRING USER CHARACTERISTICS FROM ONLINE BEHAVIOR AND CONTENT

Abstract

by

Munira Syed

Content consumption and generation is a major part of the Internet experience. Product and service-providers strive to improve user experience through personalization of services, recommendations, and understanding user interests. For this purpose, inferring user characteristics, such as demographic information, from their behavior, would help understand their preferences. Through this dissertation, we show that by using content and behavior data, we can characterize users for the purpose of improving their experience through personalization in the domains of learning analytics, a sub-domain of education, and online content consumption. We discuss two challenges: (1). representing users given heterogeneous, industry-scale volume of data, and (2). improving the representation of underrepresented groups of users, which is the imbalanced classification problem.

CONTENTS

Figures vii

Tables ix

Acknowledgments xi

Chapter 1: Introduction 1
1.1 Overview of Domains 2
1.1.1 Learning Analytics 2
1.1.2 Online News Consumption 3
1.2 Challenges 3
1.2.1 User Representation 3
1.2.2 Underrepresented Users 5
1.3 Exploratory Data Analysis 6

Chapter 2: Introduction to Learning Analytics 10

Chapter 3: ABCs of MOOCs: Affect, Behavior, and Cognition 12
3.1 Overview 12
3.2 Introduction 13
3.3 Course Structure / Study Design 14
3.4 Findings / Affect 17
3.4.1 Emotion Quadrants and Trajectories 17
3.4.2 Emotion Quadrant Transition Likelihoods 20
3.5 Inter-Play of Affect, Behavior, Cognition 24
3.5.1 Behavior 24
3.5.2 Cognition 26
3.5.3 Co-occurrence Analysis 27
3.6 Contributions 29

Chapter 4: Implicit and Explicit Emotions in MOOCs 32
4.1 Overview 32
4.2 Introduction 32
4.3 Related Work 34
4.4 Data Description 36

4.4.1 Course Description . . . 36
4.4.2 Explicit Emotions . . . 37
4.4.3 Implicit Emotions . . . 37
4.4.4 Combined Emotions . . . 38
4.5 Analysis . . . 40
4.5.1 Calculated Valences . . . 40
4.5.2 Implicit vs. Explicit features (RQ1) . . . 40
4.5.2.1 Feature Vectors Description . . . 41
4.5.2.2 Correlation . . . 41
4.5.2.3 Clustering of Feature Vectors . . . 42
4.5.3 Combined Sequence Features (RQ2) . . . 43
4.5.3.1 Correlation of Features with Completion . . . 43
4.5.3.2 Correlation of features with Quiz Performance . . . 45
4.5.4 Positivity Clustering (RQ2) . . . 45
4.6 Contributions . . . 45

Chapter 5: Integrated Closed-Loop Learning Analytics Scheme . . . 50
5.1 Overview . . . 50
5.2 Introduction . . . 51
5.3 Related Work . . . 52
5.4 Context and Framework . . . 54
5.4.1 Research Questions . . . 54
5.4.2 Our Framework . . . 55
5.5 Architecting for Collection . . . 56
5.5.1 Design . . . 57
5.5.1.1 Overview of the Course Design . . . 57
5.5.1.2 Assessment Design . . . 57
5.5.1.3 Standardized Grading and Gradebook . . . 58
5.5.2 Build . . . 58
5.5.3 Capture . . . 60
5.6 Analyzing for Action . . . 61
5.6.1 Identify . . . 61
5.6.2 Notify . . . 62
5.6.3 Boost . . . 63
5.7 Assessing for Improvement . . . 64
5.7.1 Evaluate . . . 64
5.7.1.1 RQ1: Identification Criteria . . . 64
5.7.1.2 RQ2: Intervention Impact . . . 72
5.7.1.3 RQ3: FYE and Overall First Semester Performance . 74
5.7.2 Report . . . 78
5.8 Discussion . . . 79
5.9 Contributions . . . 81

Chapter 6: Introduction to Online Content Consumption . . . 83

Chapter 7: Gender Prediction using Content Data . . . 85
7.1 Overview . . . 85
7.2 Introduction . . . 86
7.3 Dataset Description . . . 88
7.4 Model . . . 89
7.4.1 Steps I and II - User Representation . . . 89
7.4.2 Step III - User Representation Using Topics . . . 89
7.4.3 Step IV - Split into Training and Testing Sets . . . 91
7.4.4 Resampling . . . 91
7.4.4.1 SeqGAN . . . 91
7.5 Experiments . . . 92
7.5.1 Reuters . . . 93
7.5.2 20 Newsgroups . . . 93
7.5.3 Experiment 1: SMOTE-based Resampling . . . 95
7.5.4 Experiment 2: Text-based Resampling . . . 96
7.5.5 Using Both Resampling Techniques . . . 97
7.6 Contributions . . . 97

Chapter 8: Overcoming Data Sparsity in Predicting User Characteristics from Behavior through Graph Embeddings . . . 99
8.1 Overview . . . 99
8.2 Introduction . . . 101
8.3 Related Work . . . 103
8.4 Data Description . . . 106
8.4.1 Dataset . . . 106
8.4.2 Features . . . 107
8.4.2.1 Item-Level Features (IL) . . . 107
8.4.2.2 User Embeddings (UE) . . . 107
8.4.2.3 Content-based Features (CB) . . . 108
8.4.3 Heterogeneous Features (HG) . . . 109
8.5 Model Description . . . 109
8.5.1 Base Model . . . 110
8.5.2 User Embeddings . . . 110
8.6 Analysis . . . 111
8.6.1 Comparison of Different Feature Sets . . . 111
8.6.2 Data Sparsity Conditions . . . 112
8.6.2.1 Temporally Split Training and Testing Set Users . . . 113
8.6.2.2 Imbalanced Classification . . . 117
8.7 Predicting Subscribers . . . 118
8.7.1 Problem Formulation . . . 120
8.7.2 Data Description . . . 121
8.7.3 Experiments . . . 121
8.8 Contributions . . . 123

Chapter 9: Imbalanced Classification using Graph Embeddings 126
9.1 Overview 126
9.2 Introduction 127
9.3 Related Work 130
9.4 Problem Definition 132
9.5 Model Description 133
9.5.1 Graph Construction 133
9.5.1.1 ϵ-Neighborhood 133
9.5.1.2 K-Nearest Neighbors (knn) 134
9.5.2 Generating Embeddings 135
9.5.2.1 Random Walking 135
9.5.2.2 SkipGram 135
9.5.2.3 Negative Sampling 136
9.6 Data Description 136
9.6.1 Satimage 136
9.6.2 Events 136
9.7 Experiments 137
9.7.1 Baselines 140
9.7.2 ϵ-Neighborhood Graph Construction 140
9.7.3 KNN Graph Construction 143
9.7.3.1 Sensitivity Analysis 148
9.8 Discussion 150
9.9 Contributions 153

Chapter 10: Unified Representation of News and Social Media Content 156
10.1 Overview 156
10.2 Introduction 157
10.3 Related Work 159
10.4 Background 161
10.4.1 Named Entity Recognition 161
10.4.2 Named Entity Linking 162
10.4.3 Coreference Resolution 164
10.5 Data Collection and Preprocessing 164
10.5.1 Twitter Data Collection 165
10.5.2 TNY Data Collection 165
10.5.3 Text Preprocessing 166
10.5.4 URL Preprocessing 167
10.6 Framework 167
10.6.1 Named Entity Representation 167
10.6.2 Linked Knowledge Base 168
10.6.3 Coreference Resolution 169
10.6.4 Graph Construction 170
10.7 Graph Description 171

10.8 Article-Tweet Relatedness 173
10.8.1 Baseline - Random Matching 173
10.8.2 Evaluation 173
10.8.3 Amazon Mechanical Turk 174
10.8.4 Automatic Evaluation 175
10.8.4.1 Bilingual Evaluation Understudy (BLEU) 177
10.8.4.2 Recall-Oriented Understudy for Gisting Evaluation (ROUGE) 177
10.9 Contributions 177

Chapter 11: Conclusion 182

Bibliography 186

FIGURES

3.1 Course Structure and Segments for Analysis 15
3.2 Distribution of Reported Emotions across all Surveys 18
3.3 Learning Emotions Grouped into Quadrants using ANEW [38] 19
3.4 Trajectories of Affect Quadrants across the Course 20
3.5 Distribution of Performance Levels in the Segments 26

4.1 Histogram of Implicit, Explicit, and Combined Sequence Lengths (Sequence Length ≤ 25) 39
4.2 PCA Decomposition of Explicit (top) and Implicit (bottom) Seq. Clusters ('x': cluster centers) 48
4.3 Positivity Clustering of Combined Seqs. 49

5.1 Integrated Closed-Loop Learning Analytics Scheme 56
5.2 Platform Architecture 60
5.3 Bottom-up Method of Boosting Non-thriving Students 64
5.4 Spring 2017 - Final FYE Grades Plotted Against Cumulative GPA . 76

7.1 Classification Pipeline 89
7.2 Evaluation Results of Different Resampling Methods on Reuters and 20 Newsgroups (x axis is the ratio of minority to majority samples) . 94
7.3 Evaluation Results of Different Resampling Methods 65-80 Dataset . 95

8.1 Pipeline. The circular nodes represent users and the squares are URLs. The solid black nodes are unlabeled, whereas the gray nodes are labeled. 100
8.2 Temporal Split 106
8.3 Imbalanced Classification Comparing Baseline (Top URLs) and node2vec on a 10-Fold Cross-Validation 118

9.1 Framework 129
9.2 Different Distance Metrics on Satimage (AUROC on y-axis, % threshold on x-axis) 141
9.3 KNN Results for Satimage and Events (AUROC on y-axis, k on x-axis) 144
9.4 KNN Results for NY and CA (AUROC on y-axis, k on x-axis) 144

9.5 Degree Distribution for KNN Graphs (k plotted on the x axis, y axis shows the AUROC) 145
9.6 Sensitivity Analysis for KNN Graph Construction (y axis shows the AUROC) 149
9.7 Using Different Distance Metrics to Construct KNN Graph in Events Dataset (AUROC on y-axis) 151
9.8 Using Different Distance Metrics to Construct KNN Graph in Satimage Dataset (AUROC on y-axis) 151
10.1 Examples of Linked Entities 168
10.2 Illustration of Different Stages 170
10.3 Tripartite Graph Schema 171
10.4 Normalized Histogram of the Log Count of Entities in Tweets vs News Articles 172

TABLES

1.1 Table of Projects 9

3.1 Transition Likelihood Values at p< 0.0001 21

3.2 Co-occurrence of Quadrants, Behavior and Cognition 28

4.1 1. Number of Students vs. SAM Surveys 2. Number of Students vs. SAM Scores 38

4.2 Corr. between Implicit and Explicit Features 42

4.3 Corr. of Combined Vectors with Completion 44

4.4 Corr. of Features with Quiz Performance 44

5.1 Odds Ratio 65

5.2 Confusion Matrix for Fall 2017's Early Intervention 67

5.3 Weekly Scores Correlation with Non-thriving Students for Fall and Spring Semester 69

5.4 Improvement in FYE Grades Compared Between Students Who Do and Do Not Receive Intervention 71

5.5 Correlation of FYE Final Grades with Cum. GPA and Cumulative GPA Differences between Non-thriving and Thriving Students for Each Semester 75

5.6 Correlation of Weekly Homework with Retention 77

8.1 10-Fold Cross-Validation with Different Feature Sets and Metrics . . . 105

8.2 Age Features 112

8.3 10-fold Cross-Validation Week-Wise for Gender (Accuracy) 114

8.4 10-fold Cross-validation Week-wise for Age (mse) 114

8.5 Week-Wise Split for Age 115

8.6 Week-Wise Single-Split for Gender (Accuracy) 116

8.7 Subscription Prediction with Temporally Split Training and Testing Sets 119

8.8 Subscribers Resampling 3-Fold Cross-Validated 122

9.1 Distance Measures Used to Link Vertices in Graphs. $x = \{x_0, x_1, ..., x_n\}$ and $y = \{y_0, y_1, ..., y_n\}$ are two feature vectors represented as vertices in the graph. 134
9.2 Data Description 137
9.3 Baselines with Logistic Regression Classifier 139
9.4 Satimage ϵ-Neighborhood 141
9.5 CA ϵ-Neighborhood 142
9.6 NY ϵ-Neighborhood 143
9.7 KNN Degree Distribution 146
9.8 KNN Degree Distribution for CA and NY Only Events 147
10.1 Automatic Evaluation of Text Relatedness 176
10.2 Summary of Tools and Techniques 180
10.3 Summary of Challenges 181

ACKNOWLEDGMENTS

It takes a village to raise a child, and it takes a university, collaborators, friends, and family to raise a dissertation and graduate a doctoral student. First of all, I would like to thank my committee members, without whom this dissertation would not exist. I would especially like to thank my advisor, Dr. Nitesh Chawla, for believing in me and pushing me to reach my potential. Next, I would like to thank all my labmates, who have been a part of DIAL early during my Ph.D., new students, or my cohort, who were part of this journey with me every step of the way. Be it invaluable discussions, advice, or even solidarity regarding research, immigration, and life in general, their support made my experience better and lightened the burden during tough times. I am also thankful to all my collaborators, who allowed me to be a part of exciting and impactful projects and provided financial and technical support.

I want to dedicate this next paragraph to friends and family who could not be acknowledged elsewhere, yet were instrumental to my success. I would not be here if not for my friend, Nikhita Vedula, who encouraged me to go to graduate school by helping me with applications and rendering advice at various points in my life. My roommate turned friend Rachael Purta made me a better writer, both technical and creative, by providing valuable critique and asking leading questions. It is difficult to find a good editor, and I was lucky to find her. My friends from the department, including Aparna Bharati, Micayla Goodrum, and Pamela Bilo Thomas, provided much-needed emotional support throughout my Ph.D. I thank my friends, especially Trenton Ford and Caitlin Smith, for keeping me sane and focused during the work-

from-home and pandemic times. Stress-relief in the form of fun activities is essential for balancing work, and Julie Chaney has provided me company whenever I wanted to go for a walk, vent, or discuss novel plots. I wish to thank my family for their sacrifice and supporting my decisions no matter what, even if those decisions took me halfway across the world. And finally, I thank everyone who improved my life in every small way by offering debugging advice, moral support, a compliment, or friendship. Every little thing goes a long way in carrying me across the finish line, so thank you.

CHAPTER 1

INTRODUCTION

Ever since the invention of the World Wide Web in 1989, the Internet's permeation into everyday life has been steadily increasing due to the decreasing cost of hardware, increasing communications infrastructure, and general availability of technology [182]. We use the Internet on a wide variety of devices, access a myriad of services, and generate and consume enormous quantities of content [241, 96]. Thus, our use of the Internet leads to the production and availability of big data, which provides an opportunity to learn about the users and understand their preferences. This paves the way to web personalization, in which products and services are tailored to individual user preferences with the help of their consumption data [215]. The application domains of analyzing this consumer-related data are diverse, ranging from online education to social media and providing commercial services [239, 222]. One approach to providing personalized services to users is by centering on the users themselves. In this approach, behavior and content data are used to represent users. By understanding users' characteristics such as their emotions, performance in an online class, and demographic information relevant to the service, we can improve the personalization of the product or service for individual users. Thus, the fundamental problem being tackled in this dissertation is: *How can we leverage behavior and content data to characterize users for the purpose of improving their experience through personalization?*

While a comprehensive study of all the domains and applications of such data is infeasible, we can gain some insight into ways of learning about these users and

inferring their characteristics for personalization by focusing our attention on a few domains and challenges. We provide an overview of the domains, challenges, and finally exploratory data analysis.

1.1 Overview of Domains

We study the impact of these ideas in two domains, learning analytics and online news consumption. Both of these have very different user-bases and goals. Thus, by considering these separate domains, we hope to cover a greater variety of datasets, features, and challenges.

1.1.1 Learning Analytics

Learning Analytics is a tool used in the education domain to analyze and personalize the classroom experience of students. In this domain, the users of interest are students who take classes and whose experience we want to improve. Chapter 2 provides a brief introduction to learning analytics. Massive Open Online Courses (MOOCs) have recently become prominent for remote learning due to their low cost and high scalability, i.e., ability to cater to a large number of students compared with traditional classrooms [166]. To improve the students' educational experience, we can learn more about them from their behavior and performance. While MOOCs are accessible for online learning, in traditional education institutions, first year seminars are proliferating in colleges and universities. They are designed to help students make the best of their college experience and are considered a high impact educational practice [136, 135]. Thus, it is useful to identify students who struggle in this class and boost them to improve their overall academic performance. In both MOOCs and first year seminars, it would be beneficial to proactively identify students' characteristics, such as emotions, completion, and whether they are struggling, from their behavior and content they generate.

1.1.2 Online News Consumption

In the online news consumption domain, consumers of different demographic backgrounds such as gender, age, ethnicity, and economic status consume content on the Internet. Commercial services attempt to provide personalized recommendations, relevant newsfeeds, and targeted advertisements to consumers for an improved experience [118, 75]. Inferring users' characteristics based on their behavior would enable organizations to provide these services better, and ultimately lead to better user satisfaction due to an improved understanding of user preferences. Fortunately, the clickstream logs of users are rife with heterogeneous and multi-faceted data that has high potential. We can investigate the usefulness of the different types of features within clickstream data for demographic attribute prediction. There are many domains in which representing users and inferring their characteristics would be applicable, such as social networks, education, and content-hosting websites. Chapter 6 provides an introduction to the online content consumption domain.

1.2 Challenges

Representing users and inferring their characteristics uses a variety of features and large-scale data leading to challenges in representation. Even though the domains we explore have different data sources, features, and users, they have similar challenges. In this dissertation, we explore two challenges that are frequently encountered in both the domains.

1.2.1 User Representation

The first challenge that we consider is that of user representation. Given the heterogeneous nature of Internet data [243], different sets of features and approaches can be used to represent users and predict their attributes. For example, using behav-

ior data in the form of clickstream data generated by users, we can predict specific characteristics of the consumers, such as demographic information [75], and course completion of students in an online course [222]. In Chapter 3, we provide an analysis of students' behavior and performance in relation to their characteristics, emotions and completion. In Chapter 8, we show how clickstream data can be used to predict users' demographic information and subscription. We explore students' emotions further in Chapter 4. However, content consumed by the users can be used to accomplish the same task by generating user profiles from the content of the webpages they access[176]. In Chapter 7, we use content data to predict users' gender. We also use the content generated by students to understand their emotions better in Chapter 4. One perspective may be more beneficial than the other in certain scenarios. Investigating the advantages of different features and representation techniques may lead to useful insights about the users in the context of the problem to be solved.

Another representation challenge that cannot be solved using the above techniques is when a new user visits a website, and we do not have any behavior or content data associated with the user. In this case, social media data can be leveraged to represent them and make recommendations based on trending topics. Further, we can improve personalization by studying the relationship between the content that users consume, such as news, and social media, where users may generate their own content and express their opinions on their topic of interest. However, this task of unifying the two types of content is technically challenging due to different language formalities used on various platforms. To this end, we propose a framework in Chapter 10 that can generate a unified representation for both types of content to support downstream applications such as sentiment analysis, opinion mining, and understanding the impact of social media on journalism. Thus, behavior, content, and data external to the website, such as social media, can be used to characterize users.

1.2.2 Underrepresented Users

Another challenge of understanding users' characteristics from behavior is that some user groups are underrepresented in the user base, but are equally important. In some cases, rare users are essential for revenue purposes, such as identifying the small proportion of users who will subscribe to a service or purchase a product. More importantly, in the interest of fairness, we wish to improve the experience of underrepresented users. Mehrabi et. al. [169] define fairness in the context of decision-making as the "absence of any prejudice or favoritism toward an individual or a group based on their inherent or acquired characteristics". One of the sources of unfairness in machine learning is data. Human beings generate a large proportion of the data that is used by machine learning models. Since the distribution of users who access a particular service or participate in an activity is skewed, the data generated by these users would be biased towards the largest group of users, i.e., the majority class. When this data is used to train machine learning models, this bias is incorporated into the models. This type of bias is called population bias and exists when the distribution of the population that uses a particular service is not the same as the distribution of the target population [169]. The strategies used in imbalanced classification problems could be used to improve the representation of minority class users as well for the purpose of fairness.

In the learning analytics domain, the group of underrepresented students changes in different contexts. For example, since MOOCs have a high attrition rate, the students who complete the course are underrepresented in the data. On the other hand, in the First Year of Studies (FYS) course, which is mastery-based, most students perform very well. However, there is a small percentage of students who struggle in this class. In Chapter 5, we show a strategy for identifying these struggling students early.

In the online content consumption domain, as in most real-world scenarios, there

is an inherent demographic bias in the consumption of various types of content that leads to this imbalance problem. For example, while men have been shown to use the Internet more frequently than women, fewer male users access health or religious information online compared to women [18]. A study of various methods to tackle this imbalance problem would be key to ensuring that minority groups are not neglected and are included in the benefits of personalized services available to other users. Throughout this dissertation, we explore challenges of imbalanced groups of users in both the domains of learning analytics and online user consumption and present insights and understanding into particular problems and challenges. Inspired by these analyses, we also propose a graph-based framework for solving the traditional imbalanced classification problem in Chapter 9.

1.3 Exploratory Data Analysis

Thus, we have different domains, problems, challenges, and applications that we will examine in this dissertation. To summarize these, the following table provides a quick overview of all the datasets, problems, and methods that the rest of this dissertation comprises. As can be seen from Table 1.1, the data generated by users can be overwhelming in volume, scale, and heterogeneity. Answering the fundamental problem would require the mining of massive real-world datasets. While we have different data sources for behavior and content, within each type of data, there are many features and ways to represent the data. Given a particular task such as gender prediction, processing the data in useful ways, and identifying relevant features is not a trivial task. The set of identified features could change based on the problem. In general, the following steps are a useful guideline for exploratory analysis and understanding of the data, with a summary of challenges encountered and solutions used for individual problems given in Table 10.3:

1. Understand the problem by reading relevant literature and define the task and

organizing the data available. Domain expertise can help curtail the time spent on exploring the topic space.

2. Preprocess the data to extract relevant features. For example, text data may have to be cleaned up; samples may need to be filtered if there are missing values, etc.

3. Once a list of possible hypotheses for the data mining task has been generated (e.g., users of different demographic information have different behavior), perform statistical tests on these features to find associations with the target variable. Various statistical tests, such as correlation analysis and hypothesis testing, can be used for this step.

4. Represent the features in different ways and identify which of these methods is useful for the given task. Categorical features can be one-hot encoded, text features can be represented through bag-of-words, or dimensionality reduction methods such as embeddings can be employed to get a manageable feature vector.

5. Different machine learning models have different inductive biases, so we can try these features with different models and predict the variable of interest. For example, generalized linear models (GLMs) can be used to provide more statistical insights into the data, whereas neural networks may show the capacity of the data and features. By using interpretable models such as GLMs, Naive Bayes, Support Vector Machines, and Tree-based models, we can test the predictive power of the features and understand them with respect to the inductive biases of the algorithms.

6. Identify challenges with the data such as imbalance or noise and try different strategies to counter them, e.g. resampling data, moving average for noisy emotions. Interpretable models can help us understand the data challenges. For example, some classifiers are more robust to noise than others; thus, we can get a sense of the noise level in the data by comparing performance across different classifiers. We can also understand whether certain features are strong predictors by eliminating them, even if they are not explicitly present in the data. For example, if we want to test whether the temporal order of events is an important feature, we can shuffle the input and compare it with the model's performance when the input is ordered correctly by timestamp.

Thus, by performing the exploratory analysis outlined above, we can understand what kind of features are useful for the problem. This analysis may need to be repeated multiple times in order to refine the hypotheses and features used. Once we understand the data well in the context of the problem, we can use it to design sophis-

ticated models to solve our task. A summary of techniques used in this dissertation will be provided in Chapter 11.

This dissertation is organized as follows: The first part of the dissertation focuses on the learning analytics domain. Chapter 2 introduces this domain. Chapters 3 and 4 explore students' affect, behavior, and cognition in a MOOC. Chapter 5 focuses on underrepresented students who struggle in the First Year of Studies class. The next part of this dissertation is in the online content consumption domain, with Chapter 6 introducing it. Chapter 7 explores content-based features with a focus on imbalanced user representation, while Chapter 8 compares content-based, behavior-based, and combined features. Chapter 9 further explores graph embeddings, useful features identified from the previous chapter, for imbalanced classification. Chapter 10 provides a strategy to unify news content with social media content. Finally, in chapter 11, we conclude this dissertation.

TABLE 1.1

TABLE OF PROJECTS

Domain	Project and Research Topic	Dataset Used	Types of data used	Techniques used
learning analytics	MOOC – Study the Relationship of Students' Affect, Behavior, and Cognition	Clickstream data of 14000 students, 5027 included in final analysis	Heterogeneous time series, clickstream, and content data	Statistical Analysis, Clustering
learning analytics	First Year Experience – Identify Struggling Students, Boost, and Evaluate Interventions	Classroom data of ∼2000 students per year for three years	Time series data	Statistical Analysis
Online News Consumption and Imbalanced Classification	Gender Prediction on a Health Website by generating Content-based User Profiles	17,499 clicks from users of age 60-80 years (after filtering)	Content and clickstream data	NLP (Topic Modeling) and Supervised Learning
Online News Consumption	Gender, Age, Subscription Prediction Based on Behavior	84,380 users for gender, 64,102 users for age, and 323,809 users with 5,584,073 clicks for subscription prediction	Clickstream data	Network Representation Learning and Supervised Learning
Imbalanced Classification	Imbalanced Classification using Graph Embeddings	6,435 samples in satimage dataset and 70,533 samples in events dataset	Numerical feature vectors	Network Representation Learning and Supervised Learning
Online News Consumption	Unified Representation of Content using Entity-based Graph	25,250 News articles from The New Yorker and 3M tweets from Twitter	Social Media & Content data	NLP (Named Entity Recognition, Linking and Disambiguation, and Coreference Resolution) and Network Representation Learning

CHAPTER 2

INTRODUCTION TO LEARNING ANALYTICS

The first domain we explore in this dissertation is in the field of education. Education is imparted to students in many ways, including traditional universities and online learning platforms. In this dissertation, we take one example from each. Chapters 3 and 4 are focused on online learning, whereas Chapter 5 is focused on a classroom in a university. The use of technology in the domain of education to help and support students is a worthy cause, and the area of research that focuses on this is called Learning Analytics. In the next paragraph, we provide an introduction to this field.

Learning Analytics is a field based on technology-enhanced learning [77] that focuses on the learning process [220]. In particular, it can significantly shape and impact learning in higher education [220]. While learning analytics can be deployed on many levels (e.g. department and institution), we focus on the course-level in this work, which is concerned with learning analytics deployed in classrooms [219]. Learning Analytics has been popularly used in institutions for student success and intervention [68], with a comprehensive list of the use-cases given in Dietz-Uhler *et al.* [68].

One of the aims of Learning Analytics is helping the students succeed in their learning goals. In universities, this translates to student retention beyond the first year and persistence until graduation. At the classroom level, a metric of successful learning is their grades in the class that determine whether they pass or fail.

Massive Open Online Courses (MOOCs) have become very popular in the last

few years for online learning. Most of them are cheap, scalable, and self-paced, i.e. students can view lectures and complete different components of the course on their own timeline. Many courses on MOOCs are taught by university professors and industry professionals, which speaks to the quality of learning that can be gained from MOOCs. However, MOOCs famously suffer from a problem of attrition, where tens of thousands of students enroll for the course, but only a small proportion complete them [128, 159]. Thus, in the case of MOOCs, completing the course could be considered a primary metric of success and certification or passing the course a secondary one.

In this dissertation, we show how to use technology to support students' learning. In the MOOC study in Chapters 3 and 4, we investigate the effect of emotions on students' performance to provide them a more adaptive and personalized experience. We also further explore different methods of inferring emotions from students to support further studies and development of affective-based technology. In Chapter 5 that focuses on the traditional classroom, we assist students by identifying those who are struggling and providing them a system of intervention. We go beyond merely making the framework available to them by analyzing its effectiveness year after year.

CHAPTER 3

ABCS OF MOOCS: AFFECT, BEHAVIOR, AND COGNITION

3.1 Overview

In this chapter, we consider the domain of learning analytics and investigate students' emotions with the aim of personalizing their learning experience through providing an adaptive learning platform. Thus, we report on a study of affective states of learners in a Massive Open Online Course (MOOC) and the inter-play of Affect, Behavior, and Cognition at various stages of the course. Affect is measured through a series of self-reports from learners at strategic time posts during the period of study. Behavior is characterized in terms of a learners' engagement, interactivity, impatience and reflectivity, which constitute a set of novel high-level features derived from the clickstream of learner interactions. Cognition is evaluated from the performance of learners on assessments that are part of the course. We discover that learners in the MOOC experience multiple as well as mixed emotions as they go through the course, which we handle using the psychological dimensions of arousal and valence. This results in a set of emotional quadrants, whose co-occurrence analysis reveals a strong association with cognition and specific behavioral characteristics demonstrated by the learner. These results advance our understanding of the experience of MOOC learners to a more holistic level across the key dimensions of affect, behavior and cognition. They also have important implications for the design of the next generation MOOCs that can potentially leverage affect and behavior-aware interventions to drive greater personalization and eventually, improved learning outcomes. This chapter was published as a paper [8].

3.2 Introduction

Affect is related to cognitive, motivational and behavioral processes and is considered as an key determinant for successful learning gains [70, 199]. It is therefore crucial to have access to and ensure the emotional well-being of learners for targeted and timely feedback as well as for mitigation of affective states deemed obstructive towards learning [251]. This becomes even more critical in a self-paced learning experience as offered by traditional MOOCs. MOOCs in the present form is a typical example of a self-regulated learning model where the learner is in complete charge of the pace and strategy of learning [189]. The value in participation and performance therefore depends entirely on the motivation of the learner and the significance attributed to the content for personal goals and expectations. While it is difficult to control for people experimenting with a MOOC one can certainly aim to make the learning experience richer and more effective for learners in general. A practical strategy would be to investigate learner engagement and behavior patterns to understand the nature of interaction and overall experience. Affect forms an indispensable part of this experience and its evaluation a critical variable in the design of an adaptive and personalized MOOC learning experience.

In this work, we report on a study of affective states of learners in a MOOC and the inter-play of Affect, Behavior and Cognition at various stages of the course. The MOOC we study is an introductory course on Statistics offered on the EDX platform. We investigate and report on the following Research Questions in this chapter:

RQ1: *What affective states do learners go through while taking a MOOC? How stable are these states over time, and which transitions across states are more (or less) likely?*

RQ2: *Are there any significant relationships between a learners' reported affect, observed behavior and cognition?*

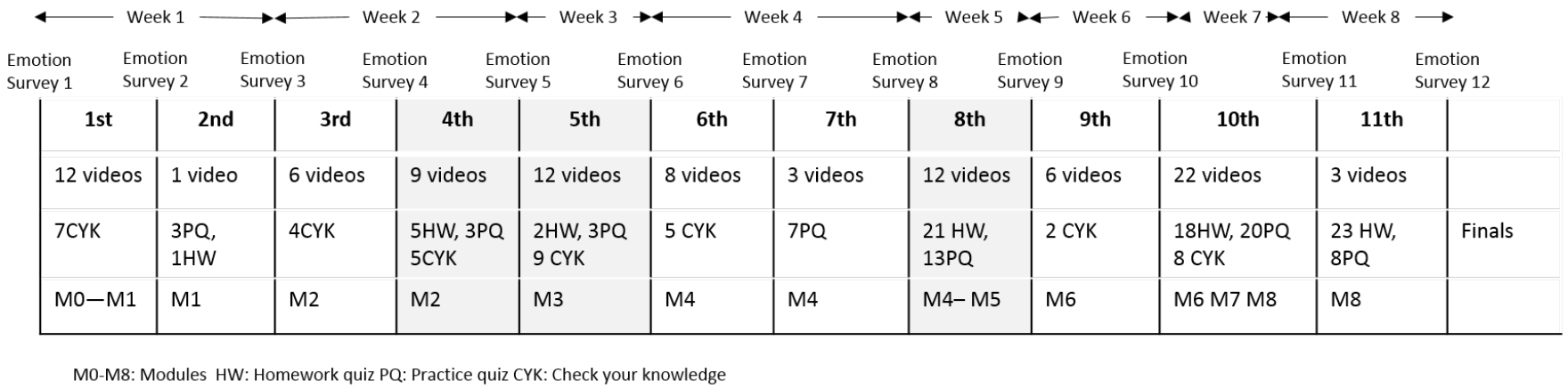
We find that learners in the MOOC experience multiple as well as mixed emotions

as they go through the course. Learners also have a higher likelihood of persisting in the same emotional state (or quadrant) across course segments than transitioning to a different state. Co-occurrence analysis reveals a strong association between the affect, observed video behaviors and the learning outcomes. Learners expressing negative emotions are associated with low performance. In terms of learner behavior, high interactivity is not necessarily associated with desirable outcomes and skipping portions of videos is not necessarily bad. Our results have significant implications for the design of the next generation MOOCs as they can provide the foundation for affect and behavior-aware interventions that drive greater adaptivity and personalization and eventually improved learning outcomes. While we see this work as novel within the space of MOOCs, there exist earlier efforts to study affect transitions [70, 168, 64, 212] and the inter-relations between affect, behavior, and learning outcomes [40, 193, 20] although mainly in the context of ITSs.

This chapter is organized as follows: Section 3.3 introduces the MOOC we investigate in this chapter and describes our study design. Section 3.4 reports on our findings on learner affective states and transitions. Section 3.5 discusses the relationships observed between affect and learner behavior and cognition. Section 3.6 concludes the chapter by outlining the key contributions, discussing current limitations and directions of future work.

3.3 Course Structure / Study Design

This work is based on an introductory course on Statistics offered on the EDX platform as a traditional MOOC. The course comprised of eight modules plus a final ninth module consisting of assessment of the overall course. The demographic information of the students was not collected during the course so the data used for analysis in this work does not contain any personally identifiable information. The MOOC had a total enrollment of 24,279 students from across 183 countries. However,



1st	2nd	3rd	4th	5th	6th	7th	8th	9th	10th	11th	
12 videos	1 video	6 videos	9 videos	12 videos	8 videos	3 videos	12 videos	6 videos	22 videos	3 videos	
7CYK	3PQ, 1HW	4CYK	5HW, 3PQ 5CYK	2HW, 3PQ 9 CYK	5 CYK	7PQ	21 HW, 13PQ	2 CYK	18HW, 20PQ 8 CYK	23 HW, 8PQ	Finals
M0—M1	M1	M2	M2	M3	M4	M4	M4– M5	M6	M6 M7 M8	M8	

M0-M8: Modules HW: Homework quiz PQ: Practice quiz CYK: Check your knowledge

Figure 3.1: Course Structure and Segments for Analysis

only about 15,000 students had any activity recorded in the beginning two weeks. There was a significant dropout of students in the initial two weeks to approximately 8000 by third week to just about 1200 students who continued in the course until module 8. In this work, we included only those students who accessed the course content in module order, participated in the self-reported emotion surveys at least once and completed the emotion self-reports in time order. This resulted in data from 5057 students (4167 dropped out before module 8, 890 completed until last module).

As part of the course design, a total of 12 periodic emotion surveys were conducted wherein students were asked to self-report on their current emotional state voluntarily. Compared to sensor-based affect detection methods, self-report is an efficient methodology for capturing the subjective emotional experience of learners, is technically easier to deploy at scale (as in a MOOC) and has high face validity. The number and placement of these surveys was designed to balance the need to collect a learners' affect data at regular periodicity while not imposing too much burden on the learner or not coming across as overly intrusive. In each survey, a student was asked to categorize their affect state by selecting at least two emotions from a pre-selected list of fifteen emotions. The list of emotions were derived from previous studies on learning-centric emotions [199] and some that were considered relevant to a MOOC setting e.g. Isolation. The final list of emotions consisted of: *anger, anxiety, boredom, confusion, contentment, disappointment, enjoyment, frustration, hope, hopelessness, isolation, pride, relief, sadness, and shame.* No specific definition of these emotions was provided to students as these are commonly used in everyday language and therefore intuitively familiar.

The 12 emotions surveys formed eleven segments of learning activities (as shown in Figure 3.1). While we included all emotion surveys to study affect distributions and their transitions, we selected only the fourth, fifth and eighth segments to study the relation between affect, behavior and cognition as these had learning activities

and emotion self-report in temporal proximity for correlating affect, behavior (video related) and cognition (from home work quiz) related information.

3.4 Findings / Affect

This section addresses the first research question RQ1 in Section 3.2 based on our findings with respect to learners' affect and its transitions over the course of the MOOC. Figure 3.2 shows the distribution of learner reported emotions aggregated across all surveys in the course. The Y axis shows % of surveys where a given emotion (on X-axis) was reported. The raw emotions reported in the surveys demonstrate the breadth of affect states experienced by learners. There is also a clear skew towards the positive affect states of Enjoyment, Hope and Contentment, followed by Relief and Pride. Amongst the reported negative emotions, Anxiety is prominent, followed by Confusion, Boredom, Frustration and Disappointment while the remaining negative emotions (Isolation, Shame, Hopelessness, Anger and Sadness) constitute a long tail. Interestingly, we observed that emotions reported together were sometimes of opposite polarity e.g. Hope and Anxiety. About 5% of responders selected more than three emotions at once.

3.4.1 Emotion Quadrants and Trajectories

To deal with the multiple emotion states and their skewed distribution in our analyses, we adopted a principled approach to group related emotions in accordance with the well-established psychological dimensions of valence (positive and negative) and arousal (activating and deactivating). Using the valence and arousal values for individual emotions from the Affective Norms of English Words [38], we grouped the self-reports into four quadrants based on positive/negative valence and high/low arousal values. The resultant quadrants and emotions they comprise are shown in Figure 3.3. Quadrant A consists of Enjoyment, Hope and Pride, Quadrant B of

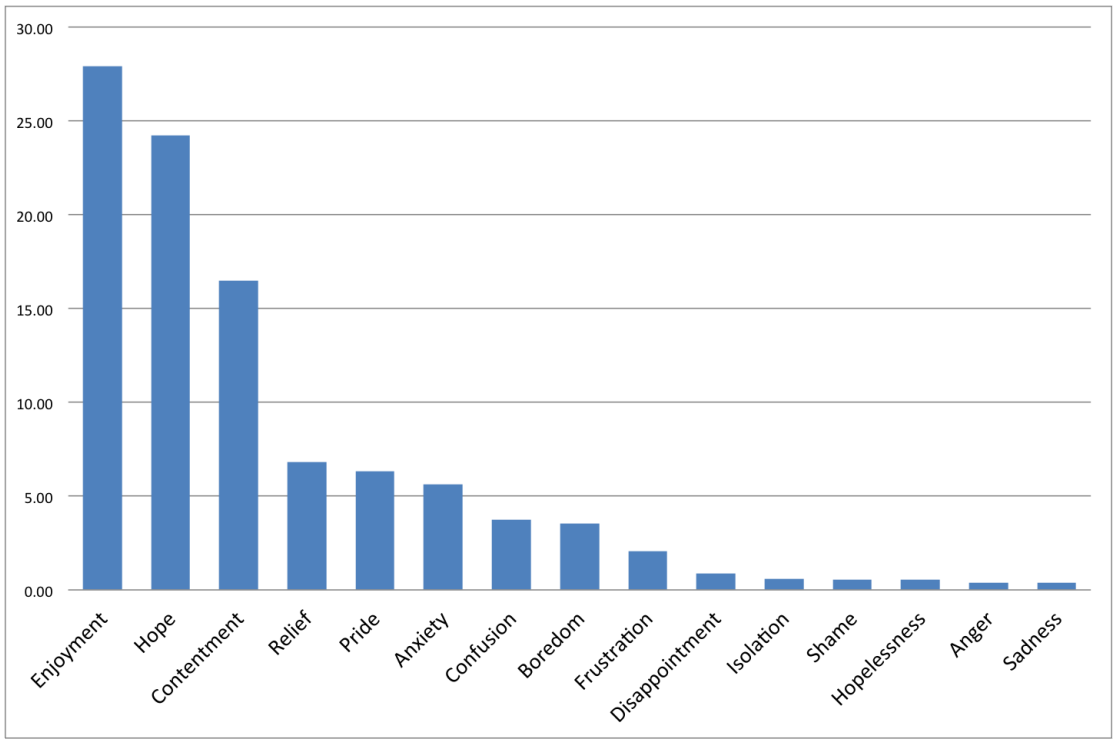


Figure 3.2. Distribution of Reported Emotions across all Surveys

Anger, Anxiety, Confusion, Frustration and Hopelessness, Quadrant C of Boredom, Disappointment, Isolation, Shame and Sadness; while Quadrant D includes Contentment and Relief.

Interestingly, this membership broadly maps onto Pekrun's categorization of academic emotions as positive activating (Quad A), negative activating (Quad B), negative deactivating (Quad C) and positive deactivating (Quad D). Only the emotions Shame and Hopelessness exchange their positions when compared with Pekrun's categorization but since the incidence of both is very low in our dataset ($< 1\%$) it does not have any significant impact on our results. So while reducing the complexity of analyzing 15 different emotions expressed by the learners in various combinations, these quadrants also have a theoretical basis in their effects and outcomes to be clustered together. Pekrun for example associates each quadrant with the use of specific

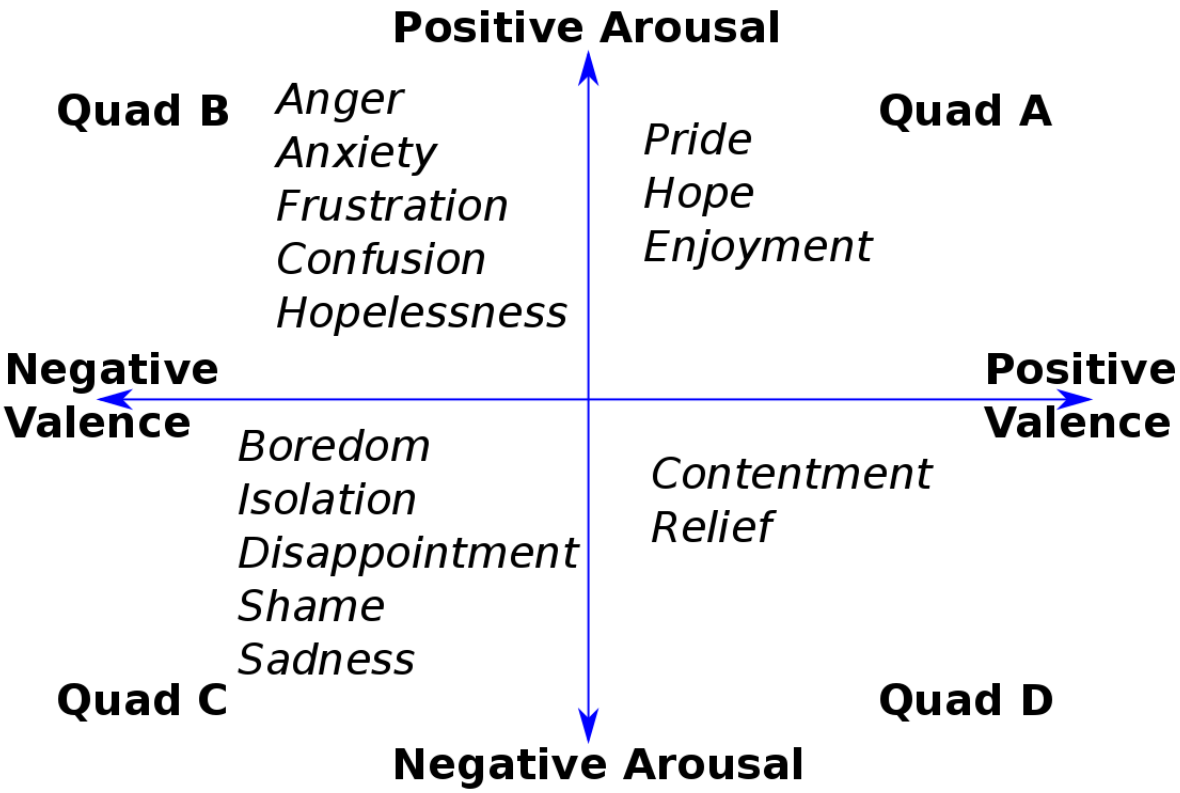


Figure 3.3. Learning Emotions Grouped into Quadrants using ANEW [38]

learning strategies and learning effects [201]. One can think of developing affect adaptation strategies on similar lines.

When learners reported more than one emotion at a time, these were often close enough to map to the same quadrant. However, at times, these were drawn from different quadrants. We grouped such self-reports according to their membership across combinatory quadrants like AB, AC, and so on along a hypothetical third dimension. The combinations of more than two quadrants had negligible occurrence ($< 1\%$) which is why we consider only Quadrants A, B, C, D, AB, BC, CD, AD, AC and BD for further analysis in the chapter.

Figure 3.4 shows the longitudinal trajectory of these emotion quadrants over

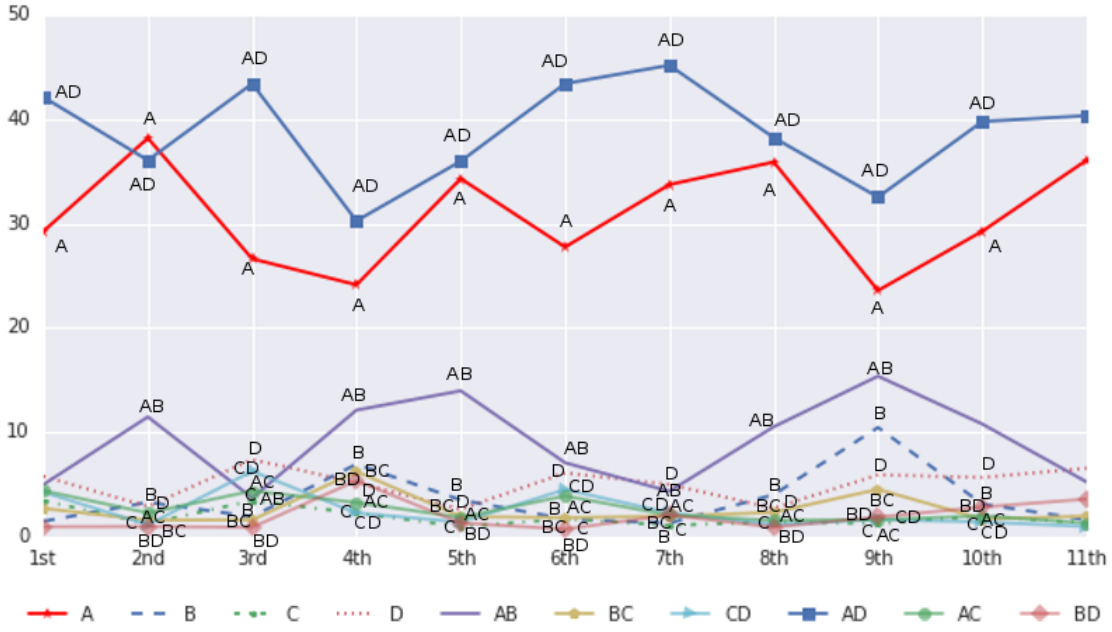


Figure 3.4. Trajectories of Affect Quadrants across the Course

the successive segments of the course, aggregated from the reports of all learners. Quadrants A, AB and AD showed the highest percentage distribution across all the surveys implying that emotions in these quadrants are most frequently occurring irrespective of the position within the course. The pre-dominance of positive emotions is similar to findings in ITS where students report Enjoyment or Flow most frequently (e.g. see [70] and [168]). The occurrence of negative states as in Quadrants B and C follow similar patterns of decreasing frequency. It should be noted that B, C and BC are interesting quadrants from the perspective of interventions during learning as they feature only negative emotions that may have a detrimental effect on learning.

3.4.2 Emotion Quadrant Transition Likelihoods

TABLE 3.1

TRANSITION LIKELIHOOD VALUES AT P< 0.0001

Quadrants	**A**	**B**	**C**	**D**	**AB**	**BC**	**CD**	**AD**	**AC**	**BD**	**Do**	**F-Score**
A	**0.27**	-0.01	0.00	-0.01	-0.02	-0.01	-0.01	**-0.16**	0.00	-0.01	0.00	44.98
B	**-0.14**	0.15	0.01	0.01	0.05	0.06	0.03	**-0.16**	0.02	0.03	0.06	8.06
C	-0.03	0.10	**0.21**	0.04	-0.03	0.13	0.04	**-0.30**	0.07	-0.01	0.09	7.65
D	**-0.12**	0.02	0.02	**0.18**	-0.01	0.00	0.03	**-0.13**	0.00	0.02	**0.12**	10.18
AB	**-0.10**	0.03	0.00	-0.01	**0.17**	0.02	0.00	**-0.16**	0.00	0.01	0.01	22.57
BC	**-0.17**	0.08	0.03	0.01	0.00	**0.19**	0.03	**-0.17**	0.01	0.04	0.07	12.24
CD	**-0.23**	0.03	0.07	0.05	-0.03	0.04	**0.20**	-0.05	0.08	0.08	0.02	7.20
AD	**-0.11**	-0.01	0.00	0.00	-0.02	-0.01	0.00	**0.24**	-0.01	0.00	-0.03	93.07
AC	-0.05	-0.01	0.05	-0.01	0.00	0.06	0.04	**-0.14**	0.12	0.05	0.01	5.95
BD	**-0.14**	0.06	0.10	0.06	0.09	0.03	0.03	0.00	0.03	0.17	0.07	2.84

Recent studies have analyzed the affective trajectories of critical learning relevant emotions like boredom, flow, confusion, frustration, delight, and surprise. These concur in their findings that learners generally tend to persist in the same affective state [64, 70, 168, 212]. Following this line of enquiry in exploring the emotional transitions of learners, we investigated the transition likelihoods of the learners in our dataset across the different quadrants. To compare with previous work, we compute D'Mello et al. [70] as the transition likelihood for affective transition analysis according to the following formula where C is Current and N is Next:

$$L(C \rightarrow N) = (Pr(N \mid C) - Pr(N)) \,/\, (1 - Pr(N)) \tag{3.1}$$

The Transition likelihood, L, computes the probability that a transition between two affective states (C$\rightarrow$ N) will occur. The formula accounts for the base frequency of the Next affect state in assessing the likelihood of a particular transition. The denominator normalizes scores between $-\infty$ and 1. Therefore, an L value equal to 1 translates to emotion Next always following the Current emotion; an L value equal to zero means the likelihood of Current transitioning to Next is equal to chance while an L value lower than zero indicates the transition to be less than chance.

Here the transitions are computed for each state for each student (total no. of transitions = 25239) and then averaged in order to find the transition likelihood from one quadrant to another. The mean values of L are then compared in a series of ANOVAs to determine whether the differences are statistically significant. The transition likelihood values for quadrants A through BD are shown in Table 3.1. The rows in Table 3.1 represent the Current affect quadrant while the columns represent the Next quadrants. Specifically, each row i of the table indicates the transition likelihoods of quadrant at i to each of the quadrants represented in columns A through to J. ANOVAs indicated significant variation among the transitions at $p < 0.0001$. Significantly meaningful transitions were determined using Tukey post hoc tests and are highlighted in Table 3.1. The main observations from the transition

analysis are:

- State-to-State transitions along the diagonal are all significant except B-B, AC-AC and BD-BD. This indicates that learners have a higher likelihood to persist in the same emotion quadrant than to transition to a different one.
- The transition to B-B is not significant as revealed through post hoc comparisons. B consists of emotions like Confusion, Frustration and Anxiety that are more spontaneous emotions and therefore may have not be as likely to persist as emotions in some of the other quadrants. At the same time, it should be noted that the transition likelihood of B to A and D that are positive quadrants is significantly below chance meaning that learners go into complex emotional orientations after B and not necessarily into a completely positive state.
- Similarly, AC-AC and BD-BD feature emotions from diagonally opposite quadrants making it unlikely to be a stable state.
- The column on Dropout shows the likelihood of each quadrant transitioning into student dropout from the course. Only D appears to be significantly related with Dropout. This is interesting because D has satisfactory but deactivating emotions like Contentment and Relief and could imply a sort of positive dropout wherein the learners have satisfactorily achieved their goals or expectations from course and hence do not continue further.
- With the exception of C and AC, the transition likelihood of any quadrant to A seems to be statistically significantly below chance. Similarly, the transition likelihood to AD from any state except CD and BD is also significantly below chance. This is interesting because both A and AD have the highest distributions across the course also supported by the statistically significant transition likelihood of A-A (0.27) and AD-AD (0.24). This seems to imply that A and AD are the most stable states and that students generally tend to be in positive emotional orientation during the course.

The results of transition likelihood in general correlate with previous research in affect transitions about the persistence of emotional states albeit at a longer time frame. Therefore the fact that learners tend to stay in a particular affect orientation across multiple interactions with learning content often spaced over weeks is an important finding. Also, our findings on transition trajectories and likelihoods of emotion quadrants are novel as opposed to transitions among individual emotions. Finally, it is worthwhile to note that this is perhaps the first formal study exploring affect transitions occurring in a self-paced learning environment as MOOCs as

against previous studies that have been conducted in a one-one setting in intelligent learning environments.

3.5 Inter-Play of Affect, Behavior, Cognition

In this section, we study the relationship between a learners' affect, behavior and cognition in the MOOC in order to address the second research question RQ2. We begin by explaining how we model a learners' behavior from raw video clickstream data and cognition from assessment performance.

3.5.1 Behavior

The interaction behavior of learners can be characterized using clickstream analysis on the digital trail of their MOOC activities. Analysis of this data can be used to study and uncover patterns of interest that may correlate with higher level categories of interest [221, 40, 130]. While most work in interaction data analysis has focused on prediction of performance and dropout rates, only recently has affect received some focus. The motivation is to explore whether certain behavioral patterns are associated with affect states so that a learner model can be built for eventual affect prediction using clickstream data. In our analysis we attempt to investigate precisely this aspect through a set of measures derived from the raw clickstream events while watching lecture videos.

For a specific video content, there can be multiple viewing sessions. The behavior metrics for a video are obtained from the behavior metrics of its individual sessions. The lecture videos are split into segments to measure the portions accessed. In our experiments, the video segments are of length 10 seconds.

Impatient: For a specific video content and video session, the impatience score of a learner is the fraction of video segments not yet watched with respect to the total number of segments in the video and considering segments watched in past sessions.

The Impatience score of a video and group of videos is the average of impatience scores across all viewing sessions and individual videos in the group respectively.

Reflective: For a specific video content and video session, the reflective score is the fraction of video segments re-watched during the current session with respect to the total number of segments in the video and considering segments watched in past sessions. The Reflective score of a video and group of videos is the average of reflective scores across all viewing sessions and videos in the group respectively.

Interactive: For a specific video content and video session, interactivity is the ratio of the total number of events generated in the session to the total session duration in seconds. All browser generated events like play, pause, stop, etc. captured in the clickstream data are considered as interactivity events. Interactivity of a video and group of videos is the average of interactivity across all viewing sessions and videos in the group respectively.

Engaged: For a specific video content and video session, engagement is measured as the ratio of viewing session duration to the length of the video content. The Engagement score of a video and group of videos is the average of engagement scores across all viewing sessions and videos in the group respectively. In the case of overall engagement score, engagement with all videos within a group is considered irrespective of being watched by the learner or not. Therefore, overall engagement score decreases for students when they do not watch all videos within a group.

The number of forward / rewind events are accounted for as part of a learners overall Interactivity. We consciously avoided using the number of fast forward events as a measure of Impatience and the number of rewind events as a measure of Reflectivity (since many learners close a session early or re-watch in a new session).

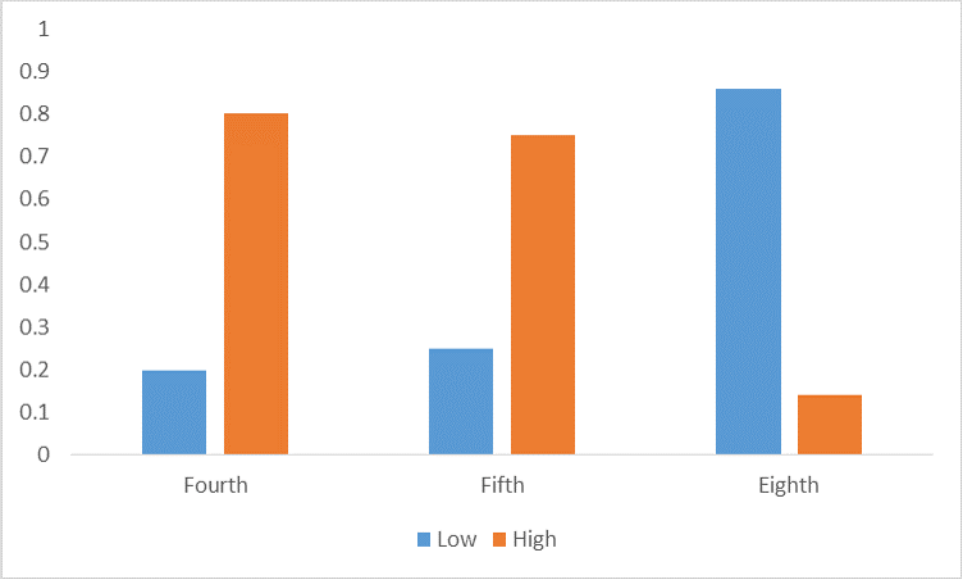


Figure 3.5. Distribution of Performance Levels in the Segments

3.5.2 Cognition

We measure cognition as an outcome of learning tested via performance on assessments. The MOOC data set had three sets of assessments: (i) check your knowledge (CYK) with multiple retry option after every video lecture, (ii) practice problems and (iii) homework problems. For the current analysis, we considered the performance scores in only the homework problems as the level of cognition achieved. In our analysis, CYK and practice problems are not considered while measuring cognition as CYK allowed multiple retries while both these assessments were not proximal to emotion surveys when compared to homework problems.

The overall performance in home work problems was above average and therefore, we chose a higher threshold for expected outcome and categorized the students having more than 75% of correct answers in the quiz as HIGH and LOW otherwise. For a group of homework assessments within a time period, similar threshold of 75% of correct answers was considered as cognition level of HIGH. Figure 3.5 shows the

distribution of performance level among the students within individual time periods. It is evident that homework problems in the 4th and 5th time periods had higher performance achievement while the 8th time period had very few students (30%) achieving the HIGH performance band.

3.5.3 Co-occurrence Analysis

We analyze the association between quadrants and the set of behavior and performance measures using the Lift metric. Lift is a data mining technique used to learn association rules by taking antecedent-consequent pairs (X, Y) and computing the support from data by taking the ratio P(X and Y) / P(X) * P(Y). Lift scores greater than 1 are considered an indication of occurrence more frequently than that expected by chance. Lift scores were computed between the emotion quadrants and the behavior and cognition features for all the three course segments (4th, 5th and 8th). Taking $p < 0.0001$, the significant associations that appear in all the three segments are shown in Table 3.2. In addition, we computed the Lift for individual emotions against the behavior and cognition metrics. These are discussed in relation with the findings of Table 3.2.

We find that learners in Quadrant A are associated with high engagement. Individually, Enjoyment within Quadrant A is associated with both high engagement and high performance. Quadrant B comprises of crucial emotions like Confusion, Frustration and Anxiety, and these show an association with low performance but also high interactivity. Quadrant C comprises of negative deactivating emotions and is probably the most undesirable state in terms of its impact on learning. Boredom, a frequent emotion within Quadrant C, shows association with both interactivity and impatience. While one would expect a strong association with low performance here, we find that this occurs only in 2 out of 3 segments. Quadrant D is associated with impatience and high performance. The combination of impatient behavior

TABLE 3.2

CO-OCCURRENCE OF QUADRANTS, BEHAVIOR AND COGNITION

Qs	Interactive	Reflective	Impatient	Engaged	Cognition
A				X	
B	X				Low
C	X				
D			X		High
AB					Low
AD		X			High

together with performance is an interesting relationship that could be examined further. Quadrant AB is a combination of individual emotions from Quadrant A and B. Our results show that Quadrant AB co-occurs with low performance. Moreover, Engagement feature associated with Quadrant A is absent in Quadrant AB. Quadrant AD, consisting of emotions from Quadrant A and D, is associated with Reflective behavior and higher performance.

There are few interesting observations from the co-occurrence analysis using the Lift metric. Learners with emotions from only the positive quadrants, D and AD, are associated with high performance. Individual emotions Hope and Pride from Quadrant A and AD are also found to be Reflective, indicating they access portions of the video lectures multiple times when compared to learners from other quadrants. Being Engaged with the content is observed in Quadrant A and this indicates learners engage with the content more often when they are in a positive emotional state. Quadrants with negative emotions (B, C and AB) are not associated with Engaged and Reflective behavior. Quadrants B and AB are associated with low per-

formance. Confusion, an individual emotion from Quadrant B is associated with low performance.

3.6 Contributions

In this chapter, we have performed an exploratory analysis of students' behavior and performance by studying their relationship with the characteristics of emotions and dropout/completion. We measured behavior using the video viewing behavior in the clickstream logs and performance through homework assessment. We defined higher-level behavioral features such as engagement, reflectivity, impatience, and interactivity using video viewing metrics.

Our investigation into RQ1 reveals that the learners experienced a wide range of emotions in course of the MOOC. While these emotions are predominantly positive in nature, learners often experience multiple emotions at the same time, and even emotions of opposite valence and/or arousal. Learners have a higher likelihood of persisting in the same emotional state across course segments than transitioning to a different state. The only exceptions seem to be states dealing with spontaneous emotions (quadrant B) or with emotions drawn from diagonally opposite quadrants. Finally, the statistical likelihood of a learner transitioning to the dominant positive quadrants (from most of the negative or mixed quadrants) seems significantly below chance. This indicates that additional interventions may need to be designed to motivate learners in sub-optimal affect states and shift their trajectories in a positive direction.

Our analyses into RQ2 reveals that there exist statistically significant relations between the affect, observed video behaviors and the learning outcomes. Learners in Quadrant A are engaged with the video lectures, and learners in Quadrant D and AD are associated with high performance. On the other hand, learners expressing negative emotions with positive arousal (emotional states in Quadrant B and AB)

are associated with low performance. Also, students in the negative affect states, residing in Quadrants B and C exhibit high interactivity. Overall, Interactivity is not necessarily good and Impatience is not necessarily bad, and these need to be understood in the context of the learners' affect orientation and other behavioral traits to determine whether there is a risk of poor performance outcome. The co-occurrence of affect quadrants with behavioral features validates our initial goal of defining conceptual metrics in order to capture the learning traits of learners based on their video watching activity.

We have also explored defined the higher-level quadrant features from individual emotions. As emotional states are complex, the quadrant representation of multiple/co-occurring emotions, as proposed here, can serve as a proxy to study and explore their relation with behavior and performance. This has the potential to design newer and simpler form of affect sensing and appropriate learning interventions when multiple emotions co-occur.

There are a few limitations in our approach that we intend to address in future. We plan to conduct a pre-course survey of registrants to better understand their motivations or goals and see how that correlates with their affect, behavior and cognition as well as their drop-out/retention behavior. The demographic/cultural characteristics of learners may also influence affect or behavior; however, this information was not collected as part of the MOOC we studied, and we intend to cover this in future.

While video lecture viewing is the predominant learning activity, students also participate in discussion forum, surveys, etc., inclusion of which may improve our analysis. Our model of cognition is currently based on a simple performance threshold that results in two categories. Going forward, we intend to explore richer models that may involve finer categorizations and investigation into the nature and complexity of assessments.

Finally, the eventual goal of this research is to provide a foundation for designing

personalized and timely interventions based on a well-informed view of a learners' affect, behavior and cognition. By designing an intervention framework for MOOC learners and evaluating its effectiveness in driving higher engagement, we aim to improve retention and performance. In Chapter 5, we describe such a framework that is used to identify struggling students and improve their performance in a first year of studies seminar.

CHAPTER 4

IMPLICIT AND EXPLICIT EMOTIONS IN MOOCS

4.1 Overview

In the last chapter, we observed that understanding the affect expressed by learners is essential for enriching the learning experience in Massive Open Online Courses (MOOCs). However, online learning environments, especially MOOCs, pose several challenges in understanding the different types of affect experienced by a learner. In this chapter, we explore this problem further by defining two categories of emotions, *explicit* emotions as those collected directly from the student through self-reported surveys, and *implicit* emotions as those inferred unobtrusively during the learning process. We also introduce positivity, as a moving average measure to study the valence reported by students chronologically, and use it to derive insights into their emotion patterns and their association with learning outcomes. We show that implicit and explicit emotions expressed by students within the context of a MOOC are independent of each other, however, they correlate better with students' behavior compared to their valence. This chapter has been published as a paper [227].

4.2 Introduction

The exploration of emotions expressed by students in Massive Open Online Courses (MOOCs) has caught the attention of researchers for improving the remote and non-contact learning experience [256, 69, 147, 55]. A few examples of these studies infer emotions of students from their behavior [147], surveys collected during the course

[69, 8], clickstream data and discussion forums [256, 55]. The relationship between students' emotions and their behavior, learning outcomes, engagement, and dropout within the MOOC context is established in [8, 246, 211].

Emotions experienced by students during a course impact their behavior and learning outcomes [101, 200]. Detecting the emotion experienced during learning is difficult, and various methods have been employed for this purpose. The methods used to sample emotions mainly fall into three categories as outlined by [252]. The first category consists of methods that take snapshots of students' emotions during the course through survey questionnaires. These methods are intrusive to the learning process and are usually self-reported and subjective in nature. The second category detects emotions during the learning process and includes methods that sample emotions non-intrusively like facial expression detection, conversations, gaze detection, and analysis of text data generated by student interactions within the course [71, 72]. The third category measures emotions after the learning process. The first two categories are relevant to our work. In [252], the methods in the second category are assumed to counteract the limitations of the methods in the first category. Therefore, in our study we use two categories of emotions to get a more complete view of students' emotional states. In this chapter, we measure explicit emotions as the emotions recorded from student's self-reported surveys and Self- Assessment Manikins (SAMs), and implicit emotions as those from the open discussion forum posts of students.

Emotions measured in association with learning seem to be short-lived and last for a few seconds to minutes [101]. Since the emotions were expressed by students in this MOOC at different, non-uniform points in time, one of the challenges of analyzing such a series is the spontaneity of emotions. As the emotions are surveyed after the end of a video or module, we only get a snapshot of the students' emotions during the course [252]. Between two consecutive surveys, a student's emotions can

not only change multiple times, but also be conflicting, as students can experience multiple emotions simultaneously [8], which could hinder a chronological analysis of the emotions. However, even if students' emotions are spontaneous and likely to be fraught with missing data, there might be a trend to their emotions over time. An approach that leverages this idea has been proposed in [67], where the positive affect experienced by an individual is averaged over a period of time while the negative reports are ignored. Inspired by this technique, we also calculate the "positivity" of students at each point of the reported emotions and derive a positivity sequence instead of an emotion sequence. This positivity sequence is expected to be more stable over time as compared to the emotion sequence.

We study the implicit and explicit emotions expressed by the MOOC students through the following research questions.

RQ1: *Are the explicit and implicit emotions expressed within a MOOC context similar? Can one be used as a proxy for the other or are both of them equally important for characterizing a student's emotional state?*

RQ2: *What do the combined (explicit plus implicit) emotional states and positivity sequences characterize about a student's learning?*

To the best of our knowledge, this is the first attempt at investigating the effect of explicit and implicit emotion categories within a MOOC context. We find that implicit and explicit emotions expressed by students are indeed different and both are necessary to characterize student emotions. We also see that combined positivity values correlate relatively well with behavior compared to their valence values.

4.3 Related Work

The comparison of self-reported metrics like emotions and performance in self-regulated learning and other educational contexts has been studied and generally found to be inconsistent with the measured reports [94, 248, 263]. While many of

these studies measure the alignment of students' achievement calibration with their actual performance [93, 248, 94], we aim to compare the self-reported emotions of students in MOOCs against the emotions we measure from their behavior in the MOOC, in the form of interactions on the discussion forum. A direct comparison of these methods with ours is infeasible because of the difference in instrumentation and methodology. However, we will compare our general observations with the trends in literature.

We use students' self-reports of emotions along with Self-Assessment Manikin (SAM) as the explicit measures of students' emotions. Self-reports are a very common way of measuring students' emotions because of their subjective nature [88]. Collecting students' emotions through surveys is easy to deploy on a large-scale and is low cost [88], which makes them favourable for use in MOOCs [8]. SAM is a non-verbal assessment technique that allows people to rate their pleasure, represented as valence in our case, on an ordinal scale [39]. SAMs have been used to measure emotion in online learning environments [56, 69].

Among the techniques available for detecting the implicitly expressed emotions of students, analyzing emotions from texts is one of the least invasive ways of detecting students' emotions [158, 213]. Using discussion forums to detect students' emotions in MOOCs is becoming prominent due to its unobtrusiveness and low instrumentation [256]. Many sentiment analysis techniques for detecting valence from text including the word-affect lexicon used in this chapter are listed in [177], and education has been noted as one of the applications of sentiment analysis. We use Warriner's [242] word-affect lexicon to calculate the valence values of words in the discussion forum records. The effectiveness of Warriner's word-affect lexicon [242] for sentiment analysis has been demonstrated for detecting sarcasm [209], finding geographical locations associated with happier tweets [91], etc. This automatic method to detect affect from discussion forum data enables a scalable way to glean implicit affect in

MOOCs from a large number of forum posts. Sentiment analysis polarity techniques were applied on discussion forum posts in [246]. In [256], a Mechanical Turk is used to obtain confusion ratings among students through simple features like counting the number of question marks to predict the level of confusion in the discussion forum posts. They also use Linguistic Inquiry and Word Count (LIWC) to consider negation words and phrases as an indicator of potential confusion, and clickstream patterns (eg. quiz-quiz-forum) as a feature for detecting confusion. Previous research on using the discussion forum to estimate student retention and performance is complicated due to a vast amount of missing and imbalanced data [33]. We also face challenges to detect implicit emotions in the midst of context-specific terms.

4.4 Data Description

4.4.1 Course Description

We use the data from the introductory course on Statistics called "I Heart Stats" for our study. This was a self-paced MOOC on the EdX platform, and the entire course content was released at the start of the course. The course had nine modules, with the ninth module being for the assessment of the overall course. During the course, students were asked to self-report their emotions and valence through emotion surveys and SAM surveys respectively. Initially 24,279 students were enrolled in the course, however, only less than 15,000 students had activity in the first two weeks. Finally, only 1,941 students completed it. Of all the students, 1,629 responded to at least one emotion or SAM survey, and participated in the discussion forum as well. Only these students have been included in the Analysis section of this work as these are the only students generating both implicit as well as explicit emotions. Note that students completing the course are likely to have longer sequence lengths. Students not interacting with the discussion forum but are still part of the course cannot be

included in the analysis leading to an overrepresentation of active users. This leads to a bias through user interaction [169].

4.4.2 Explicit Emotions

Emotion Surveys: Of all the students, 6,100 submitted 21,448 emotion surveys. During the course, 12 emotion surveys were conducted in which students self-reported their current emotional state. This was optional and students could choose multiple of a list of 15 emotions: *anger, anxiety, boredom, confusion, contentment, disappointment, enjoyment, frustration, hope, hopelessness, isolation, pride, relief, sadness, and shame*. Further details can be found in [8]. The valence values of these emotions were calculated using Warriner's lexicon [242], with a scale of 1 to 9 and 5 being neutral. We shift the scale to [-4, 4] to bring the neutral valence to 0. In the case of multiple emotions being expressed, the associated valence values were averaged to obtain one valence value per survey. Thus, the surveys have positive (0, 4], negative [-4, 0), and neutral {0} valence values.

SAM Surveys: A total of 5 SAM surveys, using a 5-point scale, were conducted in this MOOC. The SAM score represented in Table 4.1 ranges from 1 to 5 with 1 being the least and 5 being the highest state of pleasure. As the distribution of the number of students corresponding to each SAM score is normal, we convert this scale to an interval scale in the range [-4, 4] linearly. In total, 5,363 students have submitted 9,512 SAM surveys with the rest of the details shows in Table 4.1.

4.4.3 Implicit Emotions

The discussion forum is a platform that students use to interact with each other, the instructor, and teaching assistant of the MOOC. In total, 1,717 students generated 5,322 discussion forum records. The posts, comments, and replies (i.e. records) on the discussion forum are used to infer the implicit emotions of students.

TABLE 4.1

1. NUMBER OF STUDENTS VS. SAM SURVEYS 2. NUMBER OF STUDENTS VS. SAM SCORES

SAM survey	**No. of students**
1	4111
2	2815
3	1354
4	906
5	326

SAM score	**No. of students**
1	3204
2	4355
3	1557
4	295
5	101

We use Warriner's word-affect lexicon [242] to calculate the valence values of discussion form records. The tokenized words in tweets are used to calculate the mean valence value of the tweet using Warriner's word-affect lexicon. We use a similar approach to calculate valence values for discussion forum records using the following steps: (i) Tokenize the records to get a list of words, (ii) Remove the stop words from the list, (iii) Make a list v of valence values associated with a word using the lexicon, if present, after re-scaling them between [-4, 4], (iv) Multiply the valence values of words/phrases that follow a negative word with -1 (eg. not, never), and (iv) Return the average valence value of list v.

4.4.4 Combined Emotions

Throughout the course, students have multiple opportunities, explicit or implicit, to express their emotions. The 12 emotion surveys, 5 SAM surveys, and valence values calculated from discussion forum records were interleaved and ordered chronologically

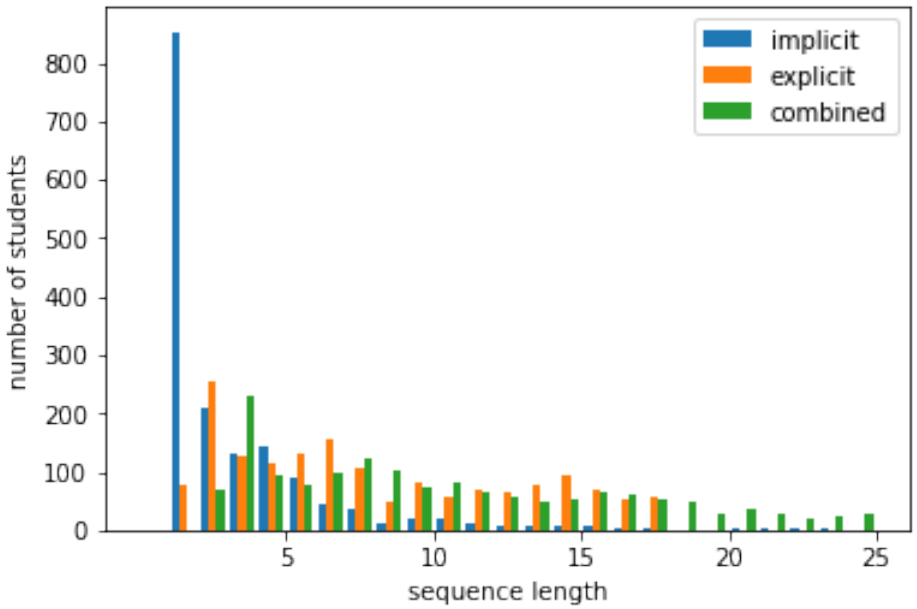


Figure 4.1. Histogram of Implicit, Explicit, and Combined Sequence Lengths (Sequence Length $\leq$ 25)

for each student to form a combined sequence of valence values.

A histogram of the number of reports corresponding to the number of students in Figure 4.4.4 shows that the highest number of students (14%) has a maximum combined sequence length of 3 with the number of students tapering down after that point. The maximum number of reports corresponding to a student is 74, as this student was very active in the discussion forum.

To mitigate the spontaneous nature of emotions, we calculate the positivity of students at each report from the valence sequence values. Thus, if a student reports one negative emotion among a string of positive emotions, the impact of the negative emotion is reduced because of the previously expressed positive emotions. We define positivity as follows.

Positivity: Let $r_1, r_2, ..., r_n$ be the reports made by a student until element n such that:

$timestamp(r_{i-1}) < timestamp(r_i)$ for all i. The valences are normalized between [-1, 1], instead of [-4, 4], by dividing them by 4. Let $p_1, p_2, ..., p_m$ be the positive normalized valences where $m <= n$ and $m + 1 > n$. The positivity at the nth element is given by $(p_1 + p_2 + ... + p_m)/n$.

In other words, an element of the positivity sequence is calculated by averaging over only the positive valences in the sequence until that element. Since students have reported more positive than negative valences both explicitly and implicitly, calculating negativity instead of positivity would lead to extremely sparse sequences.

4.5 Analysis

4.5.1 Calculated Valences

Section 4.4.3 lists the steps to calculate the valence values of the discussion forum records. To validate these valence values, 440 samples of the discussion forum records were manually annotated by three human raters in which each rater chooses one, two, or none of the 15 emotion choices that students had for their emotion surveys. The fourth rater is the calculated valence. We use Fleiss' Kappa [22] to calculate the inter-rater agreement by converting the valence scores to positive, negative, or zero valence. The inter-rater agreement of the three human raters is 0.457 (moderate agreement), whereas the inter-rater agreement of the four raters including the calculated valences is 0.218 (fair agreement) [236]. While the agreement including the calculated valences is lower, it is adequate, and so we use the calculated valence of these discussion forum records as the implicit valence values.

4.5.2 Implicit vs. Explicit features (RQ1)

Both implicit and explicit sequences are instances of irregular time-series data. However, since emotion data is spontaneous and might change multiple times between

consecutive reports [101], averaging, downsampling, interpolating or duplicating valence values in an emotion sequence might misrepresent the true emotional trajectory of the student.

4.5.2.1 Feature Vectors Description

Since the valence sequences are not uniform in length, we create fixed length feature vectors for analysis. The features are used in Sections 4.5.2.2 and 4.5.2.3 with their description given: (i) *pos*: ratio of the number of positive valences to the total length of the sequence (ii) *neg*: ratio of the number of negative valences to the total length of the sequence (iii) *neu*: ratio of the number of neutral valences to the total length of the sequence (iv) *trans*: ratio of the number of transition of valences from positive to negative or vice versa in the sequence to the sequence length (v) *pos_neg*: ratio of the number of transition of valences from positive to negative to the sequence length (vi) *neg_pos*: ratio of the number of transition of valences from negative to positive to the sequence length (vii) *range*: calculated by subtracting the minimum valence value from the maximum valence value expressed (To normalize the value the resulting range is divided by 8, as the valence values lie in the range [-4, 4].) (viii) *seq_len*: length of the valence sequence (integral value).

4.5.2.2 Correlation

In Table 4.2, we see that *pos*, *neg*, and *neu*, as defined in Section 4.5.2.1, between implicit and explicit emotions of students are not correlated with each other. This shows that both types of sequences are somewhat independent of each other and might show different insights into students' affect. There are relatively few neutral discussion forum records which is why its correlation with completion is not significant. That is why transitions from neutral to positive and negative valences, and vice-versa have been left out of the features list. The sequence lengths seem to be

TABLE 4.2

CORR. BETWEEN IMPLICIT AND EXPLICIT FEATURES

Features	**Pearson's r**	**Spearman's ρ**
pos	0.0401	0.0696**
neg	0.0413*	0.102***
neu	0.0150	0.0380
seq_len	0.346***	0.422***
trans	0.125***	0.162***
neg_pos	0.113***	0.165***
pos_neg	0.0805**	0.127***
range	0.243***	0.257***

* p-val.<0.1, ** p-val.<0.05, *** p-val.<0.0001

mildly correlated showing that students reporting more emotions in the emotion surveys were also more likely to submit more records in the discussion forum. This correlation is expected since the number of students with larger sequence lengths decreases as seen from Figure 4.4.4.

4.5.2.3 Clustering of Feature Vectors

We cluster the 7-dimensional feature vector to identify groups of similar students using K-Means. To visualize the clusters created, we decompose the 7-dimensional feature vectors of students' implicit and explicit emotion sequences to a 2-dimensional space using Principal Component Analysis (PCA) separately. The PCA decomposition in Figure 4.5.2.3 shows very separable clusters in the 2-dimensional space. The explicit clusters have significantly different ratios of course completion: orange:

37.2%, purple: 25.5%, olive: 51.9%. Similarly, the completion ratios of the implicit clusters are: red: 34.5%, blue 32.6%:, green: 60.3%, with the green cluster having significantly more students completing the course than the other two.

4.5.3 Combined Sequence Features (RQ2)

From the previous subsection, we saw that implicit and explicit sequences are not identical and should both be incorporated into a student's valence trajectory. So we use both implicit and explicit sources of emotions ordered by time to generate a combined valence sequence for students. The features from Section 4.5.2.1 are used in the analysis below.

4.5.3.1 Correlation of Features with Completion

We generate the 7-dimensional feature vector from the combined valence sequence for each as defined in Section 4.5.2.1 and show the correlation of each dimension with completion in Table 4.3. Completion is defined by a student reaching module 8 [8]. We see that *seq_len* has the highest correlation with completion possibly because sequence length could act as proxy for the amount of time students spent in the course. A similar reasoning might hold for *trans*. The *pos*, *neg*, or *neu* features do not seem to be correlated with completion. However, *neg_pos* seems to be better correlated with completion than *pos_neg*. This supports our intuition that students transitioning from a negative to positive emotional state are more likely to stay in the course, compared to the other way round. The feature *range* is better correlated with completion than *trans* which indicates that higher intensity of changes in emotions is more likely to result in completion.

TABLE 4.3

CORR. OF COMBINED VECTORS WITH COMPLETION

Feature	**Pearson's r**	**Spearman's rho**
pos	-0.0549***	-0.110***
neg	0.0807***	0.156***
neu	-0.0382**	0.0828***
neg_pos	0.215***	0.300***
pos_neg	0.112***	0.201***
trans	0.186***	0.223***
seq_len	0.523***	0.460***
range	0.390***	0.392***

* p-val.<0.1, ** p-val.<0.05, *** p-val.<0.0001

TABLE 4.4

CORR. OF FEATURES WITH QUIZ PERFORMANCE

Features	**average**	**minimum**	**maximum**
range	-0.0804*	-0.181***	0.0735*
seq_len	-0.232***	0.0681*	-0.405***

* p-val.<0.1, ** p-val.<0.05, *** p-val.<0.0001

4.5.3.2 Correlation of features with Quiz Performance

The performance score of students for a quiz is normalized between 0 and 1. The average, minimum, and maximum performance score of the quizzes (total 4) that students have attempted is used as the y-variable for correlation. The features that are significantly correlated with these statistics using Pearson's correlation are in Table 4.4. While the negative correlation with *seq_len* is unsurprising given that harder quizzes are towards the end of the course, the positive correlation with *range* suggests that student who experience extreme emotions tend to perform better.

4.5.4 Positivity Clustering (RQ2)

We compare fixed length positivity sequences by clustering the first 10 elements of 767 students who have a sequence length of at least 10. We see that k=3 is the highest number that shows no overlap of cluster centers. While there is no significant difference between the clusters for quiz performance, the difference between clusters in terms of quiz participation using ANOVA is significant at p-value ¡ 0.05. Specifically, in the k=3 chart in Figure 4.5.4, there are more students in the most positive (green) cluster that do not submit a single quiz (29.3%) than the other two clusters (20%). A possible explanation is that students had trouble with the quizzes and the ones who did not attempt them were more likely to be happier. All three cluster centers converge towards a narrow range of positivity, suggesting that students tend towards the same positivity in the course even though they started out differently.

4.6 Contributions

In this chapter we further explored the emotions of students, which are an important characteristic to understanding them and personalizing their experience in MOOCs. We defined positivity as a moving average of individually reported valences

by students since emotions are noisy, spontaneous, and conflicting. We also define a fixed length feature vector to since valence sequences are irregular time series and vary in length. This enabled us to compare the implicit and explicitly generated valence sequences.

Similar to the studies [94, 248, 263], we found that the self-reported emotions did not reflect the implicitly measured emotions. Clustering students by their emotion sequence had different ratios of students that completed the course in each cluster. This observation is similar to what [94] found about different learning strategies and activity of students. To investigate whether the temporally proximal self-report was correlated with the outcome completion, we measured the correlation of the last reported valence and the final positivity in the students' sequences with completion. However, similar to [263], we found no correlation. This suggests that the proximity of students' emotions to the outcome completion does not have a bearing on completion.

Through RQ1, we show that both the implicit and explicit emotion sequences are independent of each other and contribute different emotional information. Through RQ2, we showed that students tend to converge towards the same positivity even though they start out differently, indicating that they end up feeling the same way. This might be because of external factors that remained constant for all the students, e.g., how the course was conducted, possibly explaining the lack of correlation with the course outcomes. We see significant differences between these clusters in quiz participation but not in other learning outcomes. This may be because students who did not attempt the quizzes did not struggle through the course and remained relatively happy. Our results show that there is potential for identifying different groups of students that participate in a MOOC.

Table 4.2 shows that the explicit and implicit sequences are associated with behavior, but not valence. One of the possible reasons is that students who participate more in the discussion forum tend to submit more surveys as well but the two types

of sequences do not corroborate each other in valence. From Table 4.3, we also observe that students who feel negatively about the course and then transition to a positive emotional state are more likely to stay in the course. We found that the range of valence that students experience is more indicative of their course completion and quiz performance possibly because the students who struggle through the course report higher valence values after achieving their course objectives, resulting in their highly varied emotions.

We have also used content in the form of discussion forum records as a source of implicitly expressed emotions. To compare all the disparate sources of emotions, we converted the emotions expressed through different sources to the same numerical scale of valence using a word-affect lexicon. However, a limitation of our work is our sentiment analysis technique that uses a bag-of-words model with the discussion forum records only and does not consider other implicit measures of emotions. In this work, we have only relied on a single word-affect lexicon. However, we can make the calculated valence values more stable by triangulating the valences with other lexicons. In the future, we hope to improve our sentiment analysis so as to capture more nuanced implicit emotions.

We would also like to improve granularity and quantify the extra information conveyed by either type of emotion sequence. Even so, as most emotion research in MOOC relies on only one category of emotions, we conclude that it might be advantageous for researchers in this area to supplement their current method with a method from the other category of emotions. It is important to continue exploring emotions in MOOCs in pursuit of goals such as the personalization of MOOCs, improving the emotional well-being of students, and the design of MOOCs.

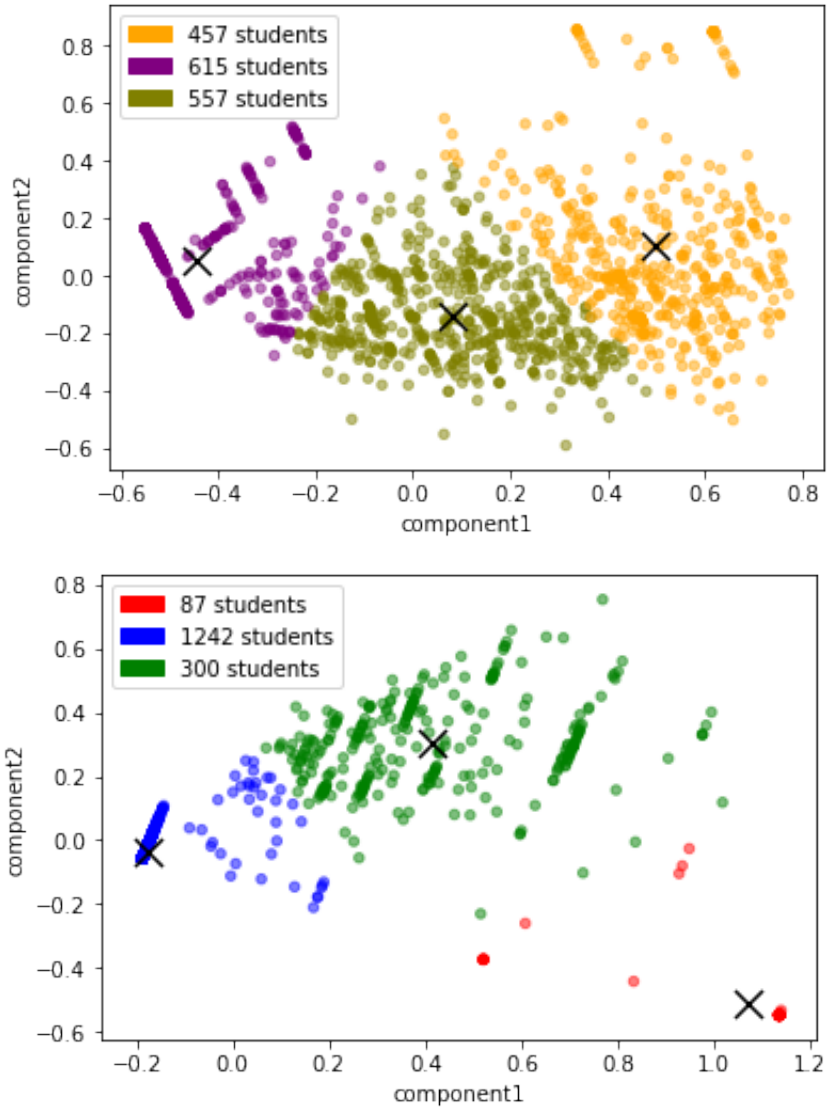


Figure 4.2. PCA Decomposition of Explicit (top) and Implicit (bottom) Seq. Clusters ('x': cluster centers)

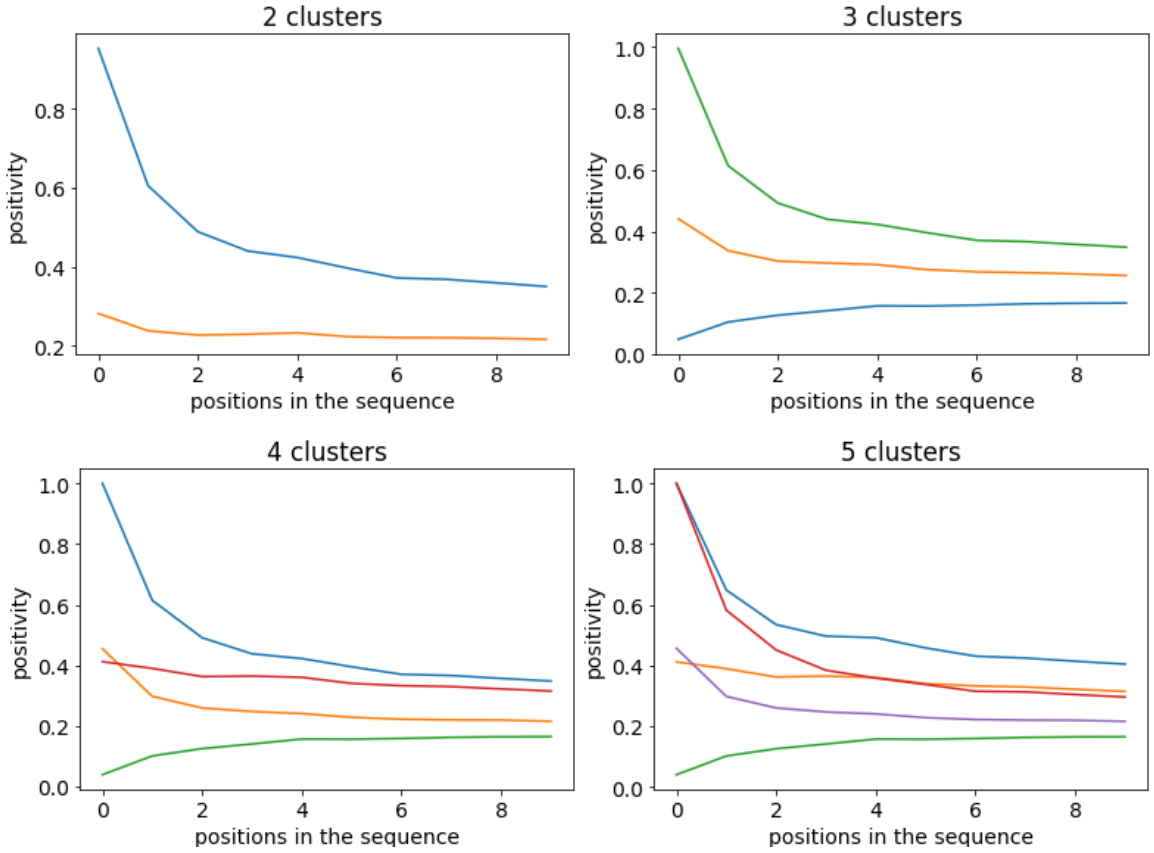


Figure 4.3. Positivity Clustering of Combined Seqs.

CHAPTER 5

INTEGRATED CLOSED-LOOP LEARNING ANALYTICS SCHEME

5.1 Overview

In this chapter [1], we move to the traditional classroom setting, in which students and instructors are face-to-face within a bounded physical space. We choose to examine a specific class, First Year Experience (FYE), that all incoming freshmen at the University of Notre Dame are required to take for 2 credits. While many classes on campus exist, the FYE course has the largest class size and can support more analysis. Interestingly, the grades of students in this class are correlated with their overall GPA. This provides us with a strong motive to help students succeed in this class. Identifying non-thriving students and intervening to boost them are two processes that recent literature suggests should be more tightly integrated. We perform this integration over six semesters in an FYE course with the aim of boosting student success, by using an integrated closed-loop learning analytics scheme that consists of multiple steps broken into three main phases, as follows: Architecting for Collection (steps: design, build, capture), Analyzing for Action (steps: identify, notify, boost), and Assessing for Improvement (steps: evaluate, report). We close the loop by allowing later steps to inform earlier ones in real-time during a semester and iteratively year to year, thereby improving the course from data-driven insights. This process depends on the purposeful design of an integrated learning environment

[1]We thank everyone in our team for the three years of this study, especially, Kevin Abbott, Kevin Barry, Chris Clark, Hugh Page, Maureen Dawson, Patrick Miller, Sharif Nijim, and Erin Hoffmann Harding.

that facilitates data collection, storage, and analysis. Methods for evaluating the effectiveness of our analytics-based student interventions show that our criterion for identifying non-thriving students was satisfactory and that non-thriving students demonstrated more substantial changes from mid-term to final course grades than already-thriving students. Lastly, we make a case for using early performance in the FYE as an indicator of overall performance and retention of first-year students. This chapter has been published as a paper [226].

5.2 Introduction

Identifying at-risk students is an established area of research in learning analytics [14, 178, 250, 183], whereas an emerging area explores the design of learning analytics interventions [249]. There is not much research, however, that attempts to combine the two and close the learning analytics loop. Furthermore, to the best of our knowledge, existing studies do not examine the evolution and evaluation of intervention mechanisms over the years when a course is offered multiple times. A possible reason for the lack of such studies is the problem of designing an infrastructure that will make this analysis possible. In this chapter, we aim to show how the combination of learning data, platform design infrastructure, identification of non-thriving students, and intervention can give us actionable insights on students who show signs of potentially struggling in the course and beyond, early in a semester.

University of Notre Dame, is a medium-sized (a total of 8,530 undergraduate students by Fall 2016) private institution located in the Midwest U.S. The overall student body is 53% male and 47% female with 98% of students who began their studies in Fall 2015 returning in Fall 2016.

The 98% first-year retention rate makes it difficult to identify students who are not thriving at this university. However, the creation of a new First Year Experience (FYE) course in 2015 presented us with an opportunity to explore potential solutions.

This course, now in its fourth year, is mandatory for all first-year students, of which over 1500 students are included in our analyses, and draws 125+ instructors, each of whom leads a standardized section of no more than 19 students. Consisting of two semester-long courses each worth one credit-hour and associated with a letter grade, the FYE helps students make a meaningful transition to collegiate life by integrating their academic, co-curricular, and residential experiences. As a mastery-based course, FYE is designed with the expectation that all students who put in the necessary effort should not only succeed but also be on a pathway to thrive. As a result, on an average, 90.00% of the students get an A as their final grade, with a standard deviation of 1.39% every semester.

In this chapter, we demonstrate that our FYE course can give us insights into overall retention and the performance of students including all the classes they enroll in. Since the analyses include the majority of the first-year student body, subtle student behavior patterns that might often be overlooked in smaller classes can be more apparent. We also discuss our data pipeline process for capturing and analyzing all the data, our techniques to identify students who are not thriving in this introductory course, and our attempts at boosting them.

5.3 Related Work

Recently, there is a growing emphasis on closing the learning analytics loop [62, 192, 165, 78] in which the results of predictive analytics and insights gleaned from them are used to improve the current or next iteration of a course in the form of interventions [62] and learning design [179]. In particular, Clow [62] recommends a five-step approach to this closed loop cycle: Capture, Report, Predict, Act, and Refine. We show in this chapter how we use historical classroom data to improve our identification of non-thriving students in the next iteration of the course, thus closing the learning analytics loop. A recent example of this effort is by Choi *et al.* [58] who

identify at-risk students using a simple metric and provide interventions to those students in one small course. In our work, we perform identification and intervention on the entire first-year body of students and repeat it for several semesters.

Every learning platform/institute has its own data collection and storage systems, and attempts to standardize these have not been widely successful [65]. In response, we propose a framework that can be tailored to build the underlying infrastructure. We aim to offer both reproducible steps that can be implemented in any classroom setting and to provide our work as evidence for the successful deployment of these cyclic steps.

First-year seminars, including our FYE course, have become increasingly popular. They are a high impact educational practice [136, 135] and their significance for retention, persistence, and engagement has been shown in the literature [178, 97, 205, 129]. Thus, we find it important to help students thrive in our FYE course. To generate the rich data on students' course activity in FYE (which is essential for actionable learning insights) and to promote active and student-centered learning, we took a flipped classroom approach [32, 35]. In our flipped FYE course, students participate actively in seminar-style discussions which build on their preparatory work at home.

The first step of identifying students that need to be boosted, the non-thriving students of a course, has been a popular area of research in the learning analytics community [14, 178, 250, 183]. Different data sources like demographic data, students' performance, and behavior are used to predict at-risk students. Some of these studies show improvements in student's grades after deploying these systems [14]. But, it is not clear if the improvement in learning outcomes is because of the intervention provided or if there were other factors involved because of a lack of evidence [78]. While these studies focus on at-risk students, we find the use of this term misleading in our case and potentially harmful as these students are not necessarily at risk of

failing the class, but may struggle later or in other aspects of their campus life. In other words, our aim is not to help students survive, but to ensure that they thrive. We do this by including not only students at a risk of failing the course but also those in the bottom 2% of the course grades.

Once the non-thriving students are identified, various intervention strategies can be employed to improve the performance of these students. Some intervention strategies shift the effort to the students, with the system sending them an email [14], whereas other intervention mechanisms include intensive intervention within or outside the classroom [90]. Another commonly used approach is providing feedback to students using dashboards [194, 63, 216]. Our intervention strategy involves the campus support system in the form of academic advisors to directly intervene with the students, aided by diagnostic gradebook reports, to help identify and solve the problems that the students might be facing. The relationship between academic advising and student retention has been shown in [225, 235, 108]. In a later iteration of the course, we added a personalized action plan via email intervention. The progress of students can be monitored either at the end of the course [14] or throughout the course [90]. In our work, we tracked the progress of students at multiple points before the end of the semester, intervening regularly at mid-term and, in some semesters, earlier as well.

5.4 Context and Framework

5.4.1 Research Questions

Because the course is designed to provide a consistent environment for all students, it has easily accessible data and a controlled environment for research. In order to investigate the effectiveness of our approach, we outline research questions that help organize our analysis and evaluation of the strategy from multiple perspectives:

RQ1 (Identification Criteria): *How do we identify students who are not thriving and offer them support and encouragement to boost their success?*

RQ2 (Intervention Impact): *What is the impact of our early and mid-semester analytics-based boost?*

RQ3 (FYE and Overall First Semester Performance): *If the FYE is a common course for all students, could it serve as an indicator of overall first-year performance and retention?*

5.4.2 Our Framework

We organize this chapter according to our integrated Closed-loop Learning Analytics Scheme (iCLAS) shown in Figure 5.1. Section 5.5 (architecting for collection) describes the architecture of our system with "design", "build", and "capture" as its steps of actions. Section 5.6 (analyzing for action) describes our identification and intervention loop with "identify", "notify", and "boost" as its steps of actions. Section 5.7 (assessing for improvement) describes the effectiveness of the various components of our scheme with "evaluate" and "report" as its steps of actions.

The loop intersects in "evaluate" and "identify" due to our commitment to continuously improve our ability to identify and boost non-thriving students. We also close the loop between "report" and "design" by reporting our findings to the design team, so that they may implement the required changes in the next iteration of the course. This iCLAS process should create a coherent and cohesive workflow that transcends courses and stakeholder perspective. Through this exposition, we hope that all stakeholders (program directors, instructors, advisors, students, data/learning scientists, and platform engineers) can recognize the design value of our integrative approach.

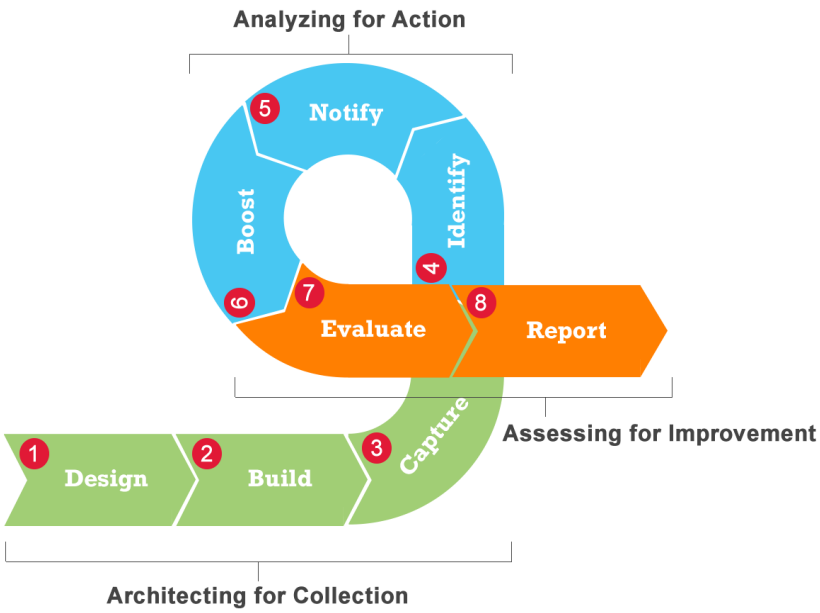


Figure 5.1. Integrated Closed-Loop Learning Analytics Scheme

5.5 Architecting for Collection

The first three steps in this foundational phase (Figure 5.1) optimize the opportunity for learning analytics by designing an engaging learning experience with standardized assessment and building a Next Generation Digital Learning Environment (NGDLE) that captures multidimensional data. In the first step, we *(1) design* an active, integrative student-centered learning experience for the course. With mastery learning in mind, we wanted the course to encourage critical, independent thinking for our students. In the next step, we *(2) build* a standard and integrated learning environment for the course. We wanted the environment to follow the NGDLE interoperability with integrative analytics, advising, and learning assessment principles in mind [42, 11]. Lastly, we ensure that our architecture has the capability to *(3) capture* student data from multiple sources in real-time into a centralized learning record warehouse as shown in Figure 5.2. With a centralized warehouse, we wanted

to empower key stakeholders to make decisions based on actionable reports using real-time multidimensional data. These three steps ensure our ability to perform comprehensive analysis for action and conduct continuous assessment for improvement.

5.5.1 Design

Our primary design goal was to deliver an engaging and consistent learning experience to all FYE students and capture multidimensional data they generated in a centralized location in real-time to fuel actionable learning analytics. The goal was accomplished through an iterative and incremental development approach. The following section describes the approach and solutions in detail.

5.5.1.1 Overview of the Course Design

Students meet in FYE sections for 50 minutes over 13 weeks of each semester. Before each session, students are provided with online materials to review and reflect on in a written weekly prompt assignment due before each class. In-person class meetings are discussion-based or active experiential learning on campus. After class, the weekly prompts are scored and students begin the process of preparing for the following week. At the mid- and end-point of each semester, students are given an in-class participation grade. Major assignments (integrations) occur twice a semester, at the mid- and end-points.

5.5.1.2 Assessment Design

To ensure consistency, all 100+ course sections of FYE shared the same assessments and rubrics that consisted of weekly pre-class assignments, major integration assignments and participation grades both at the middle and end of the semester. Thus even though each section was graded by its own instructor, the students' grades

were standardized and comparable across all the sections. Prior to each week's class, students were expected to complete a short reading/video viewing and write a 200-word response to a preparation prompt related to that material. The rubric had only 3 levels to create a simple and low-stakes scoring system for instructors to evaluate if students showed reasonable preparation (20 points), partial preparation (10 points), or no preparation (0 points). Prompts were designed primarily to hold students accountable for completion of the reading/viewing and prepare them to participate in discussions during the in-class meeting. Participation scores were assigned twice a semester. By providing participation scores at mid-term, students received feedback on their level of participation and could make a change, if necessary, for the second half of the semester. Multimedia ePortfolio assignments (integrations) were submitted three times a semester in the academic year of 2015-16, then reduced to twice a semester starting in Fall 2016. The same rubrics were used to assign scores in these categories to all students.

5.5.1.3 Standardized Grading and Gradebook

Every graded item was scored from a universally-applied rubric by the instructor of the section. FYE program directors designed rubrics for weekly prompts, integrations, and participation as explained in Section 5.5.1.2. The use of common course grade-scales and identically constructed gradebook tools resulted in our ability to readily aggregate grade data from all sections and make direct comparisons and analyses across the entire first-year cohort of students.

5.5.2 Build

Our course design requires a standard and integrated learning environment. In order to implement such a learning environment, we followed the principles of ND-GLE which focuses on bridging the gaps between current learning management tools

and a digital learning environment that could meet the changing needs of higher education [42]. Sakai was chosen as the main hub for this learning environment, and we integrated all the tools required for course activities by following the interoperability and integration dimensions of NGDLEs [42]. Our integration process is iterative. We started from basic HTML iframe embeddings of videos and Google Docs onto the course webpage and upgraded to advanced vendor-provided application programming interfaces (APIs). We eventually evolved to build Learning Tool Interoperability (LTI) solutions. LTI is a standard developed by the IMS Global Learning Consortium and aims to deliver a single framework for integrating any LMS product with any learning application [5]. The LTI integration not only allows students to perform all the required tasks in one central place but also enables the secure and trusted data flow between tools.

Another critical dimension of our learning environment is "analytics, advising, and learning assessment" [42, 11]. We intentionally built the environment to internally collect various sources of tool data like grades, click data, and ePortfolio assignment text. Our attempt to continuously improve the data collection process was iterative as well. We started with manually extracting clickstream and grades data in a batch periodically and upgraded to a Learning Record Warehouse (LRW) in real-time. The LRW was implemented based on the Apereo open-source learning record warehouse solution [13].

This upgrade removes the limitation of delayed on-time identification and assistance for non-thriving students presented in the batch process. In this upgraded system, every time a student performs a task in our learning environment, an xAPI or Caliper statement describing that experience is reported and stored in the LRW. For example, an experience is written as "student A performed action B with outcome C (in context D) at time E". xAPI is a new specification for learning technology that makes it possible to collect data about the wide range of experiences a person

has (online and offline) [3]; Caliper offers the same ability with a richer set of specifications ("metric profile") [1]. Subsequently, the use of LRW solution resulted in the ability to record traces of student learning activity seamlessly in real-time. This eliminates the effort to manually extract data from each individual tool and makes real-time analytics possible. More importantly, this ambient data collection process does not impose any extra requirements on students.

5.5.3 Capture

Figure 5.2 describes the data collection process and pipeline. With the implementation of NGDLE and LRW, course activity data such as logging in and out, clicking on resources, attempting and submitting assignments were captured from Sakai in real-time in LRW. Time-on-task data such as the amount of time students actually spent on watching course videos were also collected from Panopto. This data revealed different aspects of students video watching behavior: how many times students viewed any given video, what segments of the video students selected to view, where did they stop viewing, what their average view rate was. Additionally, student performance indicators, such as their weekly prompts scores, ePortfolio integration scores, and class participation scores, were collected directly from the Sakai grades database to ensure data integrity and accuracy. This multidimensional data was

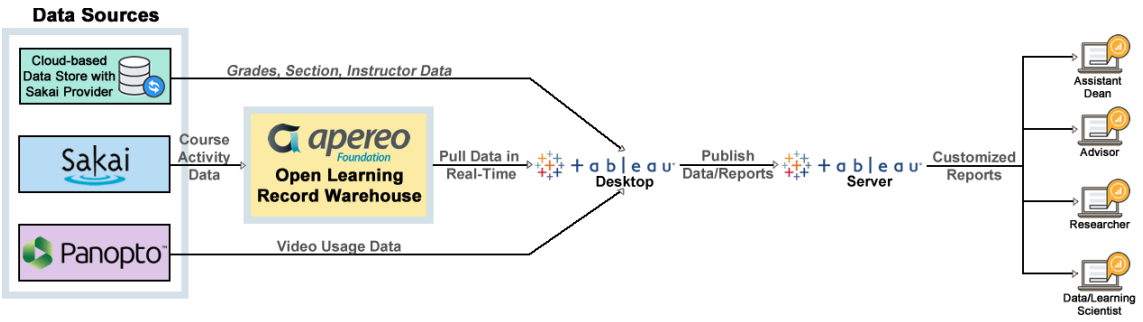


Figure 5.2: Platform Architecture

merged in Tableau to develop insightful reports. Another reason we used Tableau is it makes it easier to share raw data or reports with different stakeholders. The data collection process and pipeline shown in Figure 5.2 is essential in our research and effort to boost every student's potential to thrive.

5.6 Analyzing for Action

From the first semester of FYE, we took steps to boost students towards positive educational outcomes. The three steps on this mindful action phase (Figure 5.1) are to identify students, notify them for action, and boost their success. We *(4) identify* students using a combination of learning design predictions (an educated guess of factors that show signs of students who are not thriving) and retroactive statistical analysis of students' data that have been captured in the previous phase. Once we identified the students, we *(5) notify* them through two methods: bottom-up (inform and empower students via personalized action plan) and top-down (alert and empower advisors via one-on-one communication). In this process, we worked hard to prevent negative labeling of our students by not using words such as "at-risk" and "intervention." Instead, we adapted positive words such as "optimize", "boost", and "thrive." This leads us to the next step: to *(6) boost* the students' success. We ensure that our boost from the student action plan and advising interactions is personalized based on an individual student's circumstances. Ultimately, these three steps are designed to encourage student success in a more compassionate way.

5.6.1 Identify

In the first semester of Fall 2015, we identified two types of non-thriving behavior that resulted in early and mid-term boosts. The early boost was provided for students who had scores of 0 on their weekly prompts in weeks 2 and 3. The mid-term boost was for students who earned C- or lower at mid-term (Week 8 of 15 week

semester) based on the institutional standard cutoff for Mid-Semester Deficiency Grade Reporting [6].

In the next three semesters (Spring 2016, Fall 2016, Spring 2017), we made two significant changes. First, we only identified the mid-term boost because we were struggling with the grade data reliability and data processing efficiency for analysis. Second, we adjusted our criterion for non-thriving to B- based on the grades distribution we observed in Fall 2015. Because most students get an A as their final grade ($90 \pm 1.39\%$ on average), domain experts decided that a B- cutoff was more appropriate for identifying non-thriving students.

The data processing and reliability bottleneck was ameliorated by the implementation of the LRW in Spring 2017, which resulted in the automatic real-time data update [174]. With this improvement, an opportunity for an earlier identification of students who needed boost was presented to us. Therefore, we hypothesized (based on domain expertise) that students should be given an early boost when they showed no preparation (0 points) at least twice, either by not submitting weekly prompts or by submitting inadequate work, on assignments in between weeks 1 to 6 for Fall 2017.

5.6.2 Notify

The list of non-thriving students identified in the above section and grounds for their inclusion were shared with the FYE program director. For Fall 2015, the FYE program director notified the instructor of record in week 4 for the early boost. At mid-term, the FYE program director notified the first year advisors in week 8.

We changed our "notify" strategy over time to accommodate the requests and convenience of various stakeholders and incorporate the findings of the analysis in the previous semesters. In the next three semesters (Spring 2016, Fall 2016, Spring 2017), the FYE program director only notified advisors of non-thriving students at mid-term

due to reasons including feedback from instructors and other stakeholders regarding the feasibility of this early intervention given instructors' workload. During the Fall 2017 semester, we added the early boost back and addressed the earlier concerns by empowering students to take direct action instead of relying on instructors and academic advisors to intervene. These non-thriving students received a message from the FYE program director to let them know that their behavior might be showing signs of struggling, to ensure that they had knowledge of resources, and to encourage them to choose a personalized action plan. We also notified the FYE program director and the instructor of record regarding these students.

5.6.3 Boost

In our first semester, instructors were encouraged to have conversations with students who were identified as part of the early boost. In week 8, first year advisors conversed with students who were identified at mid-term boost. The boost action for the next three semesters (Spring 2016, Fall 2016, Spring 2017) was solely data-driven discussion between students and their first-year advisors informed by diagnostic gradebook reports.

During Fall 2017, we re-packaged our early boost based on our analysis and the availability of all the necessary technology to implement it. We asked students to fill out a short qualtrics survey with tree-based logic to help them reflect on the reasons for their lack of preparation as reflected in their grades. Each reason led to a carefully selected list of recommended actions as shown in Figure 5.3, from which a student was asked to select their personalized action plan and avoid this situation in the future. The boost action for students at mid-term stayed the same as the other semesters.

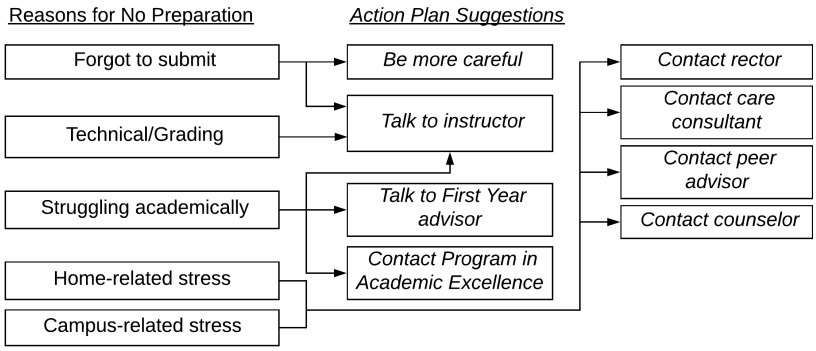


Figure 5.3. Bottom-up Method of Boosting Non-thriving Students

5.7 Assessing for Improvement

The last two steps in this continuous improvement phase (Figure 5.1) are to evaluate the impact of the course and report the findings to all stakeholders. Once we have acted on the students who were on the boosting list using the steps explained in Section 5.6, we *(7) evaluate* the intervention impact. Finally, through the architecture described in Section 5.5, we are able to *(8) report* insights into our data to multiple stakeholders using Tableau visualization. Administrators, instructors, advisors, and researchers benefited from the availability of reports on this data in order to analyze the trends of student engagement in the course.

5.7.1 Evaluate

Now that we have explained the course design and boost intervention strategy, we answer the research questions enumerated in Section 5.4.1.

5.7.1.1 RQ1: Identification Criteria

Based on domain expertise, we used the midterm grade as a ground truth for non-

TABLE 5.1

ODDS RATIO

Semester	No preparation	Non-thriving	Thriving	Odds ratio
Fall 2015	≥ 2	4	44	10.4
	< 2	15	1718	
Spring 2016	≥ 2	17	38	27.0
	< 2	28	1687	
Fall 2016	≥ 2	10	27	27.5
	< 2	20	1483	
Spring 2017	≥ 2	8	29	27.2
	< 2	15	1480	

p-value < 0.005

thriving students. Intuitively, earlier boosts would help students solve the challenges they face earlier and thrive sooner. However, identification based on assignment scores has a bottleneck of scores being available only after instructors have graded the assignments and uploaded the scores. Therefore, we hypothesized that students who showed no preparation on their assignments (weekly prompts) at least twice within the early period of six weeks should be identified as non-thriving students and need to be boosted.

To verify this hypothesis, we retroactively analyzed the semesters of Fall 2015, Spring 2016, Fall 2016, and Spring 2017 to check if showing no preparation at least twice increased the risk of students having a B- (or C- in Fall 2015) or lower grade by mid-term (week 8 of 15). The criterion (no preparation) is not independent of the outcome variable (non-thriving mid-term grades) because mid-term grades are a summation of graded weekly prompts, integration, and participation scores up to week 8. In acknowledgement of this dependence and because both the criterion and the outcome are categorical variables, we use Fisher exact odds ratio test [125] to calculate the odds ratio between no preparation on at least two assignments and not-thriving. The null hypothesis is that the criterion does not affect the outcome. Table 5.1 shows the resulting contingency matrix. For each semester, the number of students under each category is listed, with the odds ratio. The null hypothesis can be rejected with a p-value < 0.005. Thus, we see that showing no preparation for at least two assignments affects the outcome of students being identified as non-thriving by mid-term.

Clickstream data could give us an even more fine-grained view of students' assignment submission patterns. Therefore, we checked for correlation between the clickstream data of students who had a non-thriving grade by mid-term, but the results were inconclusive. We also considered using integration and participation scores, but they were populated very close to the mid-term point and were not early enough

indicators for non-thriving behavior. Hence, we decided to use showing no preparation (indicative of no submission or a score of zero) on at least two assignments as an early indicator for non-thriving students.

TABLE 5.2

CONFUSION MATRIX FOR FALL 2017'S EARLY INTERVENTION

	Grade until Week 6	
No preparation	$\leq$ **B**$-$	$\geq$ **B**
≥ 2	14	17
< 2	14	1676

In the Fall of 2017, we used this criterion as an early indicator for non-thriving students since all the required technology to implement this was finally in place. To assess its effectiveness , we show a confusion matrix with students having a B- or lower by week 6 as the ground truth. We should not use their mid-term grades as the ground truth because the intervention may interfere with their grades and change the mid-term grade. Since the early intervention occurs after week 6, the grades calculated until week 6 would not be affected and can be used as ground truth. Table 5.2 shows the associated confusion matrix. Because a very small proportion of students were non-thriving, even if all the non-thriving students were wrongly identified, we would have a high accuracy. Thus, a 98.1% accuracy is misleading as a performance metric. Instead, we used Cohen's Kappa, which is commonly used for measuring inter-rater agreement. In our case, one rater, the oracle, can look into the future after the

assignments have been graded and knows the ground truth of which students have a B- or lower at week 6. The second rater sees only the past criterion of having at least two grades indicating no-preparation. We computed Cohen's Kappa to check how much the past criterion agrees with the oracle, as a measure of the effectiveness of the second rater. The calculated Cohen's Kappa score is 0.4654 which is generally accepted to show moderate agreement [236] between our criterion and the oracle. Thus, showing no preparation on at least two assignments is a moderately reasonable criterion for identifying non-thriving students.

TABLE 5.3

WEEKLY SCORES CORRELATION WITH NON-THRIVING

STUDENTS FOR FALL AND SPRING SEMESTER

Fall 2015		Fall 2016		Fall 2017		Fall Combined		Spring 2016		Spring 2017		Spring 2018		Spring Combined	
Scores	CC	Scores	CC	Scores	CC	Scores	Corr. Coeff.	Scores	CC	Scores	CC	Scores	CC	Scores	CC
Week 5	-0.165	Week 7	-0.252	Week 4	-0.344	Week 4	-0.140	Week 5	-0.317	Week 6	-0.246	Week 2	-0.238	Week 6	-0.212
Week 3	-0.123	Week 1	-0.231	Week 5	-0.259	Week 5	-0.128	Week 1	-0.312	Week 4	-0.193	Week 4	-0.206	Week 1	-0.196
Week 2	-0.0832	Week 6	-0.205	Week 7	-0.253	Week 6	-0.0812	Week 6	-0.214	Week 3	-0.191	Week 6	-0.170	Week 4	-0.192
Week 6	-0.0670	Week 5	-0.165	Week 3	-0.222	Week 3	-0.0782	Week 4	-0.184	Week 2	-0.186	Week 1	-0.169	Week 2	-0.190
Week 4	-0.0496	Week 4	-0.0929	Week 6	-0.198	Week 2	-0.0710	Week 2	-0.174	Week 5	-0.170	Week 3	-0.146	Week 5	-0.178
Week 1	-0.0146†	Week 2	-0.0873	Week 1	-0.166	Week 7	-0.0699	Week 3	-0.142	Week 7	-0.169	Week 7	-0.125	Week 3	-0.158
Week 7	-0.00161 †	Week 3	-0.0686	Week 2	-0.146	Week 1	-0.0541	Week 7	-0.0638	Week 1	-0.0510	Week 5	-0.0421 †	Week 7	-0.110

p-value < 0.05 except where † : $p - value > 0.05$ (not significant)

In order to examine the relationship between non-thriving students and their assignments scores, we studied the correlation between them for each semester individually. We also combined the Fall and Spring semesters to see dominating patterns. Table 5.3 shows the point-wise biserial correlation coefficients for the Fall and Spring semesters. We see that all weekly scores are significantly correlated in all the semesters except Fall 2015 and Spring 2018, with a p-value < 0.05. Each semester has a different ranking of the weekly assignments depending on the correlation with non-thriving students. This ranking is not consistent over the semesters. Moreover, the differences between the correlation coefficients is not drastic. This seems to imply that all the weekly prompt grades are approximately equally important for identifying the non-thriving students.

TABLE 5.4

IMPROVEMENT IN FYE GRADES COMPARED BETWEEN
STUDENTS WHO DO AND DO NOT RECEIVE INTERVENTION

Semester	Grade change intervention Mean ± Std. Dev.	Grade change no intervention Mean ± Std. Dev.	p-value	Achievement ratio intervention Mean ± Std. Dev.	Achievement ratio no intervention Mean ± Std. Dev.	p-value
Fall 2015	2.519 ± 1.056	0.0368 ± 0.275	*	1.170 ± 0.351	0.994 ± 0.0620	*
Spring 2016	0.873 ± 0.959	0.0447 ± 0.259	*	0.939 ± 0.339	0.999 ± 0.0601	–
Fall 2016	1.298 ± 1.310	0.0403 ± 0.261	*	0.997 ± 0.380	0.997 ± 0.0500	–
Spring 2017	1.318 ± 1.195	0.0183 ± 0.187	*	1.043 ± 0.325	0.997 ± 0.0414	†
Fall 2017	0.682 ± 1.137	0.0504 ± 0.201	*	0.985 ± 0.299	1.003 ± 0.0315	–
Spring 2018	1.123 ± 1.428	0.0244 ± 0.193	*	1.039 ± 0.383	0.998 ± 0.0399	–

The p-value is calculated using one-tailed Mann-Whitney U test. Legend: p-value < 0.0001 :*, p-value < 0.01 :†, p-value > 0.05 (not-significant):–

5.7.1.2 RQ2: Intervention Impact

To evaluate the effectiveness of our interventions, we compared the change in performance of students between those who were boosted (intervention group), and those who were not (control group). This is not a truly randomized control/intervention division because the intervention group consisted entirely of all the non-thriving students, and each student in the intervention group has a lower grade than the students in the control group. To measure if the change in the outcome variable is statistically significant, we used a pre-post test with paired data. Specifically, we utilized a one-tailed non-parametric pre-post test, the Mann-Whitney U test, because the the grades of students is not normally distributed, with most of the students getting full scores. Table 5.4 shows the mean and standard deviation of different groups of students. The change in grade was computed by subtracting the mid-term grade from the final grade for each student. The Mann-Whitney U test showed that the students in the intervention group had a significantly higher change in grade compared to the control group with a p-value < 0.0001. The reported means and standard deviations of the two groups showed that the difference between them is huge. Generally, our results are consistent across the semesters. Moreover, the majority of non-thriving students (73.6%-87.6%) improved their grades between mid-term and final each semester.

While these results seem encouraging, the non-intervention group has a large fraction of students with A's in them, and these students have a very small scope of improvement compared to the students in the intervention group. To reduce the mean difference in mid-term grades of these groups, we considered a smaller subset of students in the non-intervention group. In Fall 2015, only students with a C- or below were boosted, as opposed to B- and below for all the future semesters. This gave us the opportunity to use the students within the range of B- and C- as a group of students against which we can compare the performance of the intervened students. Once again, this is not a randomized control group, because the students in this

group start out with a higher grade than the students in the intervention group. We will refer to this group as the B- to C- control group henceforth. However, we can compare the change in the grade of the students between the mid-term and the end of the semester. This result can give us some indication of the effectiveness of our intervention. The number of students with a C- or less that received an intervention is similar to the number of students in our B- to C- control group. An unpaired one-tailed Mann-Whitney U test between the changes in mid-term to final grades of the two groups showed that the intervention group had a statistically significantly greater change in grade, with a p-value < 0.0001. In fact, the mean change of grade for the B- to C- control group was 0.795, with a standard deviation of 0.607, whereas the mean change of grade for the intervention group was 2.61, with a standard deviation of 1.10.

While students in the intervention group in Table 5.4 improve their grades significantly more than the original control group, the students in the control group do not have as much scope for improvement as the students in the intervention group. To mitigate this, we calculate the achievement ratio to measure the potential that a student reaches compared to the maximum possible grade they can achieve, instead of measuring the change in grade. This is calculated by:

$$\begin{aligned}\text{achievement ratio} &= \frac{\text{final grade}}{\text{max. possible grade}}, \text{ where} \\ \text{max. possible grade} &= \text{mid term component} \cdot \text{mid term grade} \\ &\quad + (1 - \text{mid term component}) \cdot 4\end{aligned}$$

The maximum possible grade is weighted by the mid-term component of the grade, which denotes the ratio of the contribution of the mid-term grade to the final grade. Table 5.4 shows the mean and standard deviation of the achievement ratio of the

students, both in the intervention group and control group, in each semester. The achievement ratio of the intervention group is not statistically significantly greater than the control group, except in Fall 2015 and Spring 2017. We see that the effect size is small from the reported means and standard deviations of the two groups. Since we do not have a randomized control group, we cannot know definitely whether the lack of differences we see is because the intervention does not have a long-term effect, or that there were other factors related to the grades of students (e.g. internal motivation). We also note that in some cases, the average achievement ratio is greater than 1. Many students do, in fact, get a final grade that is greater than their maximum possible grade. This anomaly comes from grade reporting errors and grades being added or modified later in the semester by instructors. While we can design systems to keep track of grades entered in real-time, ultimately, on-time correct grade entry is still in the hands of the instructors. We will consider ways to reduce this problem in the future.

We can also track the set of non-thriving students from Fall to Spring semester. The academic years of 2015-16 and 2016-17 had two mid-term interventions performed in each year, with one in each of the Fall and Spring semesters. To evaluate whether students who were boosted stay boosted, we looked at how many students identified in the Fall semester were again identified as non-thriving in the Spring semester. Less than 15% of the students identified as non-thriving students were identified again in the corresponding Spring semester for all the years. Thus we see very little overlap between the Fall and Spring non-thriving students.

5.7.1.3 RQ3: FYE and Overall First Semester Performance

In this section, we explore the impact of students' performance in FYE beyond the scope of FYE. Specifically, we find the relationship between students' performance in FYE and their overall performance and retention in the first-year.

TABLE 5.5

CORRELATION OF FYE FINAL GRADES WITH CUM. GPA AND CUMULATIVE GPA DIFFERENCES BETWEEN NON-THRIVING AND THRIVING STUDENTS FOR EACH SEMESTER

Semester	Pearson r	Cum. GPA	
		non-thriving students	thriving students
		Mean ± Std. Dev.	Mean ± Std. Dev.
Fall 2015	0.386	2.76 ± 0.88	3.44 ± 0.44
Spring 2016	0.408	2.83 ± 0.76	3.45 ± 0.41
Fall 2016	0.321	2.78 ± 0.60	3.49 ± 0.40
Spring 2017	0.350	2.82 ± 0.51	3.50 ± 0.39
Fall 2017	0.325	2.97 ± 0.91	3.54 ± 0.43
Spring 2018	0.250	2.85 ± 0.68	3.54 ± 0.37
p-value < 0.0001		p-value < 0.0001	

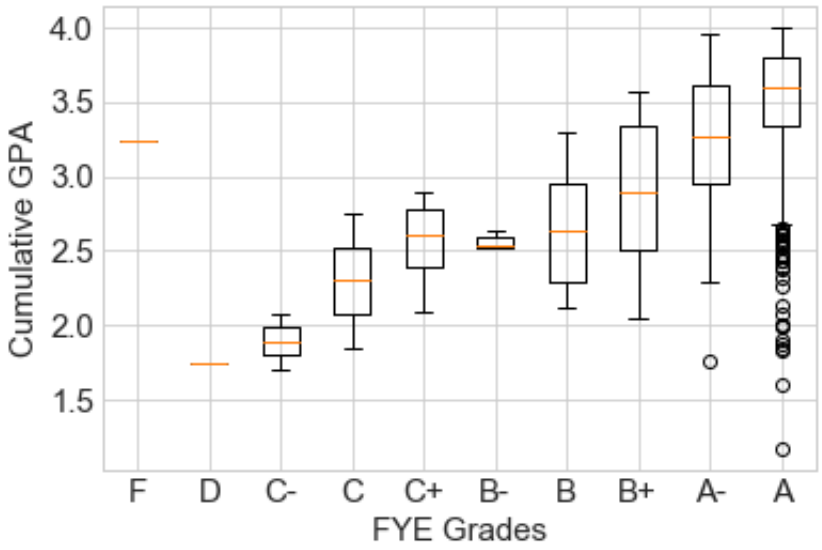


Figure 5.4. Spring 2017 - Final FYE Grades Plotted Against Cumulative GPA

Table 5.5 shows significant positive correlation between the FYE grade and cumulative GPA of students with p-values < 0.0001. For illustration purposes, Figure 5.4 shows an example from Spring 2017, where for each FYE final grade (x-axis), the boxplots of the cumulative GPA corresponding to those students is plotted (y-axis). Thus, even though the FYE is only a 1 credit course of the minimum of 12 credits a full-time student takes in a semester, we find a consistently positive correlation between the FYE grade and cumulative GPA for all the semesters. This indicates that the performance in the FYE course can provide insights into the students' overall performance.

To evaluate whether our identification criteria is effective beyond FYE, we compared the cumulative GPA of non-thriving students with those who are thriving for each semester. Table 5.5 shows the mean and standard deviation for these groups per semester. The thriving and non-thriving students have statistically significant differences in cumulative GPA every semester, with a p-value < 0.0001 using the

TABLE 5.6

CORRELATION OF WEEKLY HOMEWORK WITH RETENTION

Academic Year 2015-16		Academic Year 2017-18	
Feature	CC	Feature	CC
Week 5	-0.109	Week 1	-0.0890
Week 6	-0.0961	Week 5	-0.0814
Week 3	-0.0729	Week 3	-0.0779
Week 7	-0.0676	Week 4	-0.0733
		Week 6	-0.0727
non-thriving students	0.137	non-thriving students	0.0648

p-value < 0.01

non-parametric Mann-Whitney U test.

The issue of retention can be investigated by observing the behavior of students who are no longer with the university. The students who withdraw, are dismissed, apply for a leave of absence, or are suspended, comprise this set. By the time we identify these students within the semester, it is often too late. To identify these students earlier, we examined the correlation between the grades of students in weeks 1-7 of the Fall semester with their enrollment status in the Spring semester as the y-variable. We restricted our analysis to assignments before the intervention, because the intervention may affect the student's retention. Since the y-variable has only two values, enrolled and dismissed, we once again used point-wise biserial correlation. We do not observe significant correlations for the year of 2016-17, but some weeks are correlated with the retention of students in 2015-16 and 2017-18, shown in Table 5.6.

There is also a slight significant correlation with non-thriving students.

5.7.2 Report

The regularized and multidimensional data enabled us to perform high-level analysis and develop insightful reports to help FYE senior administrators make data-informed strategic decisions. We used Tableau, a business intelligence and analytics tool, to merge the activity and performance data from multiple sources, perform aggregate analyses, and create intuitive and insightful visualization reports. The final reports were shared with different stakeholders, such as assistant deans, program administrators, advisors, and researchers, through our Tableau server. The reports were designed to offer insights on various aspects of the FYE program. For example, to facilitate the continuous improvement of course design and provision of learning materials, we built reports to show which learning materials were most engaging and what was the optimal timing for selected materials. Based on the reports, the course design team removed the reading materials that were less engaging and adjusted the video materials to the optimal length. These strategies would help improve student engagement through better course design. We also built reports to highlight the frequency of non-submissions on assignment grouped by student, assignment, and section. These reports helped program directors monitor the progress of the course and identify opportunities to stimulate student performance. Additionally, we built program-wide grade distribution reports to empower instructors to measure and adjust their own grading practices, answer student questions on whether they are graded fairly, and help advisors to develop a holistic view of their advisees' scores. All the reports were updated and shared on a weekly basis so that FYE administrators would have the most timely information on how to continuously improve the program's effectiveness, and enhance student success and satisfaction rate.

5.8 Discussion

A limitation of our evaluation is the lack of a randomized control group. To study the long-term effects of the intervention on the performance of students in FYE, we tracked the performance of the Fall intervention and control group through the Spring semester by measuring the change between their Spring mid-term and final FYE grades. However, we did not find the intervention group to have a higher change in grades compared to the control group. This may suggest that even if our intervention has short-term effects of improving students' grades, it might not translate to a long-term performance improvement. To establish the intervention as the cause for students' grades improving, we need a randomized control group of students who do not receive the intervention. However, at the time of designing the course and intervention strategy, it was deemed unethical and unfair to provide some students with extra resources and assistance while depriving others to form a control group. This is the conundrum of impactful intervention research in real-world instead of a controlled lab setting. Finding an ethical way to provide intervention for all students that appear to need a boost while providing a control group as a way to conclusively establish the intervention as the cause of students' improvement in grades will be part of our future pathway.

We initiated an early-boost in Fall 2015 limited to students who missed assignments in both weeks 2 and 3. This was a design decision as opposed to a data-driven decision because students are allowed to switch sections in week 1 of the course. When students switch sections, the grades from the previous section are not transferred automatically. Hence to ensure a more stable population, weeks 2 and 3 were chosen to be the indicator for students who needed an early-boost. With these limitations, we saw an opportunity for improvement. We will also extend the definition and identification of non-thriving students to include indicators besides grades, e.g., clicks, cumulative GPA, the trends of students' grades as opposed to absolute grades,

and other non-academic factors.

The mutually iterative relationship we developed between the course design and data collection/analysis helped us continuously improve the student learning experience and enhance our effort to help every student succeed. Learning scientists took the lead on hypothesis, feature predictions, and interventions. Blended into the process, the data scientists evaluated and refined identification and boosting methods. The goal of this process was to continually improve the analytics-based strategy by mining multiple years of historical data.

As mentioned in Section 5.5, the upgrade to LRW resulted in the ability to do real-time analytics and the capacity to do a more refined early boost for non-thriving students. In Fall 2017, we hypothesized that students should be given an early boost when they showed no preparation at least twice in weeks 1-6. This design-based decision was backed by multiple analyses as shown in Section 5.7.1.1. There were more false positives and false negatives compared to true positives (students who are on the early-boost list) in the odds ratio analysis (Table 5.2). This may be because of the needle-in-a-haystack nature of finding non-thriving students. We hope to iterate more on the identification of early-boost list students and reduce the number of false positives and false negatives in the future.

We proposed that showing no preparation at least twice on weekly prompts as a more consistent indicator of non-thriving students. This means that we might consider notifying and boosting students automatically as soon as they miss two of their assignments instead of waiting for arbitrary $1/3$ and $1/2$ semester cutoffs. However, we would also need to be cautious and sensitive. We want to enhance students' ability to succeed instead of labeling them as at-risk. We want to nudge them into successful student behaviors instead of criticizing their inability to complete their assignments. We see First Year Experience as the appropriate course to start this endeavor. We have shown that there is a correlation between students' first

semester cumulative GPA and their performance in the course in Section 5.7.1.3. It is also a standard learning experience with less variability than other courses taught on campus. Along the way, we have developed the integrated closed-loop learning analytics scheme that consists of the backend NGDLE infrastructure, data pipelines, strategies to notify and boost students, and the front-end stakeholders interface. We hope that the NDGLE infrastructure established to capture, visualize, and analyze the data can be adapted to other large credit-granting courses.

5.9 Contributions

In this chapter, we singularly focus on identifying struggling students, an underrepresented group of students in the first year of studies course, and boosted them through interventions. We show that our identification of these students and intervention strategy are effective. We examine our research endeavor using three probing research questions that deal with our goals to effectively identify and boost students who were not thriving in a timely manner.

Our first research question deals with ways to accurately identify a small proportion (2% of the total 2,000 first year class) of non-thriving students without harm and as early as possible. We accomplished this through capturing the data they generated in real-time and performing analysis from multiple perspectives. Using various statistical methods, we can see moderate correlation between non-thriving students and no preparation on at least two assignments six weeks into the semester. However, improvements can be made in the future to reduce error in classification.

Our second research question quantifies the impact of our early and mid-semester boost. Using Mann-Whitney U analysis, we see a significantly higher change in grade but not achievement ratio for the boosted students. We also see little overlap between non-thriving students identified in the fall and those identified in the spring. However, since we do not have a randomized control group, it is challenging to establish our

intervention as the cause of non-thriving students' improvement in performance. We hope to find an ethical way to do so in the future.

The third research question investigates the impact of the FYE course on students' overall First Year grade performance and retention. We see a significant positive correlation between the FYE grade and their cumulative GPA. This is consistent with the literature that shows the impact of First Year Experience courses with respect to retention, persistence, and engagement. We will continue our investigation to understand FYE's relationship to other introductory courses commonly taken in the first year and retention.

This integrated closed-loop learning analytics scheme (iCLAS) goes beyond retroactive analytics. The scheme collects digital learning data using the Next Generation Digital Learning Environment. It takes action in real-time to boost students based on both design and data-driven insights. It evaluates its impact for continuous improvement and provides reports for multiple stakeholders in real-time or between iterations. It utilizes a First Year Experience course with standardized assessment and rubrics that provide fast and frequent low-stakes weekly assignments. This enables us to provide effective ways to obtain a real-time pulse of the students and encourages them to thrive in their first year of higher education.

CHAPTER 6

INTRODUCTION TO ONLINE CONTENT CONSUMPTION

In this and subsequent chapters, we explore the domain of online content consumption. While this domain has a huge scope and includes many problems, such as recommending content to users, advertising, and personalizing content for users, we focus on the problem of predicting users' attributes through user representation. The task of representing users is not trivial because while we have access to their behavior and content, there are many ways through which we can use the behavior to represent the user. We classify these methods of representation into three categories:

1. **Content-based approach**: Represent users based on the content they consume. Since content is unstructured, we can generate user representations using techniques such as bag-of-words, topics, and document embeddings of the articles consumed by the user.
2. **User-behavior based approach**: These features leverage the clickstream data that browsers collect, such as URL information, timestamp, device information, and location.
3. **Combined approach**: These are heterogeneous features that combine different types of features, including content and item.

In this dissertation, we focus on the demographic information of users such as gender, age, and income as the attributes of interest. Demographic prediction based on browsing behavior has applications in content recommendation, targeted advertising, and personalized news feeds. In the literature, a variety of features are used for predicting users' demographic information. Browsing history can be used to glean information about unknown users. In general, there are three types of browsing information used to predict various user attributes: click features like the number of clicks

and timestamp of clicks [204, 75], item-level features like the products [75], URLs, and items users view [118], and content-based features like the text of articles [126] and search queries [34]. There are two kinds of content-based features: the content generated by users [259] and the content consumed [204]. Users' reviews [190], [233] and tweets [82, 175, 16] are examples of the content generated by users, and these are quite commonly used for user attribute prediction. However, only 8% of the users actually create content (blogs) on the Internet [118]. Thus, content consumption-based features cover a much larger user-base whose demographics can be predicted.

While a previous study proposes a Bayesian framework for predicting a user's age and gender [118], most of the other works focus mainly on user representation while using well-established models for prediction. For example, one study uses ϵ-SVR as their predictive model and proposes a way of aggregating webpage level TF-IDF to represent users [126]. Another study investigates different types of features, including category, topic, time, and sequence, while using an SVM as their predictive model [204]. A third study proposes building a semantic user profile while evaluating them with logistic regression [176]. Similarly, one of the papers investigates different feature sets, including temporal, product categories, and some advanced features, while evaluating them with Random Forests, SVMs, and Bayesian Networks [75].

In Chapter 7, we explore a content-based method of representation through which we generate user profiles. In Chapter 8, we investigate a few user-behavior based approaches as well as combined approaches.

CHAPTER 7

GENDER PREDICTION USING CONTENT DATA

7.1 Overview

In this chapter [1], we focus on the problem of gender prediction by representing users through content-based profiles. The content-provider for the data used in this analysis is a heath website. Since topics of interest in health vary for different demographic groups, the content-based representation is potentially useful for gender prediction. Another challenge in this data is the imbalanced classification problem, since health websites typically have a skewed distribution of male and female users. We tackle this problem by proposing an oversampling technique that works in conjunction with user-profiles.

Generative Adversarial Networks (GANs) have enabled researchers to achieve groundbreaking results on generating synthetic images. While GANs have been heavily used for generating synthetic image data, there is limited work on using GANs for synthetically resampling the minority class, particularly for text data. In this chapter, we utilize Sequential Generative Adversarial Networks (SeqGAN) for creating synthetic user profiles from text data. The text data consists of articles that the users have read that are representative of the minority class. Our goal is to improve the predictive power of supervised learning algorithms for the gender prediction problem, using articles consumed by the user from a large health-based website as our data source. Our study shows that by creating synthetic user profiles for the minority

[1] We thank Trenton Ford for helpful discussions. This work was supported in part by the National Science Foundation (NSF) Grant IIS-1447795.

class with SeqGANs and passing in the resampled training data to an XGBoost classifier, we achieve a gain of 2% in AUROC, as well as a 3% gain in both F1-Score and AUPR for gender prediction when compared to SMOTE. This is promising for the use of GANs in the application of text resampling. This chapter has been published as a paper [228].

7.2 Introduction

On health websites, people focus different topics depending on their interest and relevance to them. To make predictions about individual topic interests, Nigam et. al. [187] observed and collected users' health-seeking behavior, i.e., user demographics, temporal features, and socio-economic community variables. Similarly, in our analysis, we use topic features derived from the text data of the articles read by users. By only using the content, there is a potential to generalize and create user profiles across website platforms and other domains. However, using bag-of-words representation leads to high dimensionality so instead, we use topic modeling to represent the user profiles as topic profiles.

In our data set, there is a gender imbalance problem because women tend to search and read more health-based articles online than men. In addition, the preferences and health seeking behavior of females is very different from male users [187]. Furthermore, online article content is generated and expires quickly, so learning article-specific content does not generalize well. In our analysis, we use learned topics as features to mitigate this short-lived nature of articles, with the added benefit of topics being generalizable and transferable to other domains. By concatenating all of the articles a user reads, we can build a user profile. This representation of users would be beneficial because user interests do not change as quickly as the content they consume on a website.

While most websites can have varying distributions of demographic representa-

tion, it is necessary to understand how content is consumed and interest varies based on the variety of demographic features [187]. Since we want to be able to identify the reading/consumption patterns of these under-represented users accurately, we use resampling techniques that can better represent the minority class. Imbalance can be tackled at the data level through various techniques such as oversampling (data augmentation), where we duplicate some of the minority samples, and undersampling, where we discard some of the majority samples. Undersampling techniques have the drawback of losing potentially valuable data whereas random oversampling may lead to a higher weight for the minority samples [79]. To mitigate the bias from duplicating the minority samples, Synthetic Minority Oversampling Technique (SMOTE) was introduced by Chawla et. al. [50] for generating synthetic samples of the minority class. Other SMOTE variants have also been proposed since then [79].

Most of the previously mentioned popular resampling techniques exist for resampling real, continuous data. However, when this is applied to numerical representations of text data, it could lead to the generation of noisy samples. For example, in a bag-of-words representation of text where the text samples are represented by counts of words in the vocabulary, synthetic resampling methods could generate non-integral number of words. Thus, to avoid the percolation of noise from the numerical representation of text, we can resample the minority text data using synthetic text generation techniques. LSTMs and RNNs have been used for generating text in various applications such as generating lyrics [206] and fake reviews [23]. Adversarial methods such as SeqGANs are similar to these techniques in that they use RNNs in their generator for generating text data [258]. Generative adversarial networks have been successfully used for generating synthetic samples of the minority class to augment the training set [141]. Zhu et al. focused on solving a class imbalance problem with GANs in the domain of emotion classification using images with relative success [264]. In the text domain, Anand et. al [12] used text-GANs to generate

synthetic URLs for phishing detection. While these techniques exist for synthetic text generation, their application for the task of resampling minority text data for classification, and specifically the use of GANs to do so, is under-explored. Existing resampling methods for text classification either rely on bag-of-words through term weighting [123] or generating synthetic text data using probabilistic topic models [52]. In fact, Sun et. al. [224] systematically explored the effect of popular resampling strategies on tf-idf represented imbalanced text classification problems, and found that in most cases basic SVM performs better without resampling. Our goal therefore, is to explore the use of SeqGANs for generating text data for imbalanced binary classification. For this, we propose a pipeline that represents users with user profiles and topic modeling.

7.3 Dataset Description

The browsing data used in this chapter was generated on a health-based website which collects users' demographic information from their subscribers and receives the browsing activity of these users. The data was collected from user clicks on articles from 2006 through 2015. Over this time frame, data from 263 topics related to health was collected [187]. The content of the URLs accessed by users was crawled from the website and processed by removing stop words. We experimented with a varying number of topics and decided to use 200 topics uniformly in all of our experiments. Since different age groups have varying topic interests [187] we split the dataset by age groups: 18-24 (32% of the data), 25-34 (33.3%), 35-44 (21.6%), 45-54 (14.4%), 55-64 (11.6%), and 65-80 (5.6%). For our experiments, we used the age group of 65-80 because they are at higher risk for health issues. This portion is small enough to avoid scalability issues with SeqGAN. There are 17,499 users in this age group with 13,021 females and 4,478 males (25.59% of users are male). We also discovered a long right tailed distribution with a steadily decreasing number of users as the number of

article clicks per user increases.

7.4 Model

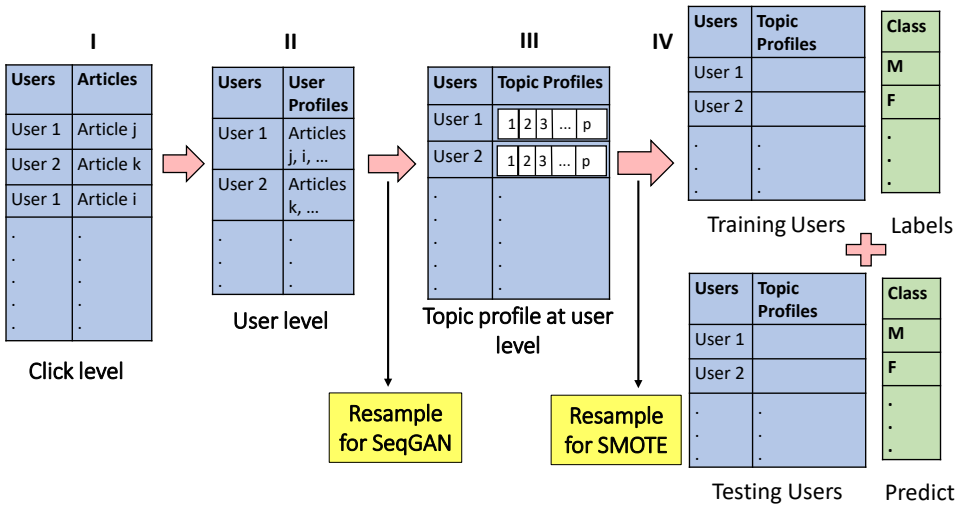


Figure 7.1: Classification Pipeline

7.4.1 Steps I and II - User Representation

Users read various articles, which is the input to the model (click level representation, Figure 7.1). At the user level, we create user profiles by concatenating all the articles read by the individual to generate a single text document correspond to each user.

7.4.2 Step III - User Representation Using Topics

We next represent a user in a structured format using topic modeling (topic profile at user level, Figure 7.1). A topic model is trained on the corpus of the individual

articles accessed by all of the users in the training and testing sets. While many topic modeling techniques such as SVD, LDA and their many variants exist, we use NMF because it is well-suited for the task of topic modeling and relies on matrix factorization. In our case, the vocabulary of the corpus is huge even after filtering stopwords. Thus, a bag-of-words representation would be infeasible for representing each user. The NMF topic model [198] is trained on all of the individual articles that appear in the corpus. The topic representation for each user is generated by transforming their profiles into the topic space.

Let the corpus D consist of articles $d_1, d_2, ..., d_m$. Each document d_j is represented by a vector of w words. Thus, the document matrix $\mathbf{D}$ has the dimensions $m \times w$. We generate an NMF topic model in the topic space of p dimensions by decomposing matrix $\mathbf{D}$ into factors $\mathbf{W}$ and $\mathbf{H}$. Thus, $\mathbf{D} = \mathbf{WH}$, where $\mathbf{W}$ has the dimensions $m \times p$ and $\mathbf{H}$ has the dimensions $p \times w$. Here $\mathbf{W}$ can be interpreted as representation of documents in the topic space and $\mathbf{H}$ is the representation of topics in the word space. NMF optimizes the objective function

$$\frac{1}{2} ||\mathbf{D} - \mathbf{WH}||_F^2 = \sum_{i=1}^{n} \sum_{j=1}^{m} \Big(D_{ij} - (WH)_{ij} \Big)^2 \tag{7.1}$$

where $\mathbf{D} = \mathbf{WH}$ and $\mathbf{W}$ and $\mathbf{H}$ are minimized alternately. A user U_i is a linear combination of all the documents in D. We represent the user by concatenating the articles read by the user. Thus, U_i is given by $\sum_{j=1}^{m} (c_j d_j)$ where c_j is the number of times user i read article d_j. The topic representation $\mathbf{W'}$ of a matrix of n users $\mathbf{U}$ (dimensions $n \times m$) consisting of $U_1, U_2, ..., U_n$, whether training or testing, would be given by $\mathbf{U} = \mathbf{W'H}$, where $\mathbf{W'}$ is obtained by minimizing the same objective function in equation 7.1 while $\mathbf{H}$ is kept constant.

7.4.3 Step IV - Split into Training and Testing Sets

The topic representation of the s users in the training set W'_{train} is the input to train a classifier, and those of the $n - s$ users in the testing set given by W'_{test} are used to predict the gender of the users as the output.

7.4.4 Resampling

In the case of the minority class, we use two resampling approaches for improving the performance of the classifier (i.e. SMOTE-based and text-based).

The data can be resampled after Step III in which topic profiles of the users are generated. This is where we will apply SMOTE and its variants. However, the text-based resampling would occur after Step II by generating synthetic user profiles of the minority class from the text of real users of the minority class. The intuition is that by applying resampling at an earlier stage, we avoid biases introduced through the conversion of text to numerical representation. Resampling at this stage can be done by Random Oversampling of the minority texts (ROS), Random Undersampling of the majority texts (RUS), and SeqGAN to reduce the imbalance between the classes.

7.4.4.1 SeqGAN

We formulate the sequence generation problem for gender classification as shown below to produce a sequence of tokens $X_{1:T} = (x_1, x_2, ..., x_T)$, $x_T \in Y$ where Y is the vocabulary of the set of candidate tokens. We train a Discriminator model D in order to guide a Generator model G. The discriminator's goal is to predict how likely a sequence X_{1T} is to be from the real sequence information. G is then updated by a policy gradient from the expected reward received from D. The formulation for the

policy gradient is shown in equation 7.2 below:

$$J(\Theta) = E[R_T|s_0] = \sum_{y_1 \in Y} G_\Theta(y_1|s_0) \times Q_{D_\Theta}^{G_\Theta}(s_0, y_1) \tag{7.2}$$

"where R_T is the reward for a complete sequence and $Q_{D_\Theta}^{G_\Theta}(s, a)$ is the expected cumulative reward starting from state s taking an action a following policy G_Θ" [258].

7.5 Experiments

Before evaluating the effectiveness of using SeqGAN resampling for the task of gender prediction, we performed some experiments on the popular text categorization datasets of Reuters-21578[2] and 20 Newsgroups[3]. These datasets have been used in comparative studies to demonstrate resampling methods on text data [224]. While Reuters texts can have multiple categories, the texts in 20 Newsgroups each fall under only one of the 20 categories. There are many ways to formulate the classification task here as an imbalanced text classification problem. However, we convert the problem into a binary classification task using one-vs-rest, where one class is taken as the minority and all the other classes are grouped together into the majority class. For numerical representation of the text data, we use NMF with 200 topics in all the experiments with the datasets to keep the representation method constant across the datasets and experiments. For a fair comparison across all resampling methods, we use the same resampling ratio for all the methods. We used the simple Logistic Regression for classification due to its interpretability while still yielding good performance.

[2] `https://archive.ics.uci.edu/ml/datasets/reuters-21578+text+categorization+collection`

[3] `http://qwone.com/~jason/20Newsgroups/`

7.5.1 Reuters

The Reuters dataset is available through NLTK [160] and we use a single-split evaluation. The training set has 7,769 samples while the testing set has 3,019 samples. We only use the categories with the highest number of texts associated with them, i.e. earn (2,877 samples in training set and 1,087 in testing set) and acq (1,650 instances in the training set and 719 in the testing set) the smaller categories lead to extreme imbalance ratios. The baselines on non-resampled data for each of the categories is the first point in each plot. In Reuters, we use F1 score taken at a threshold of 0.5 as the performance metric. The AUROC (Area Under Receiver-Operator Characteristics) and AUPR (Area Under Precision-Recall Curve) values for most of the experiments on these datasets is almost 1, and thus we use F1 score to compare them.

7.5.2 20 Newsgroups

The 20 Newsgroups dataset is available through sklearn and the training and test sets are marked in it [198], which we use for splitting the corpus for training and testing. The total number of samples in the training set is 11,314, whereas the testing set has 7,532 samples. Similar to Reuters, we use the categories rec.motorcycles (598 for training and 398 for testing), rec.sport.hocky (600 for training and 399 for testing), and sci.crypt (595 for training and 396 for testing) because these categories have a high number of articles in the dataset. In spite of that, the imbalance is greater in these sets compared to Reuters. In 20 Newsgroups, we use AUPR instead of AUROC for the reason that AUROC scores in these experiments are almost 1.

In this set of experiments, we compare the performance of resampling the minority class instances before generating numerical representations (NMF vectors) with the methods that rely on resampling the numerical representation. In the bottom row of Figure 7.2, the first point of the plots shows the performance on the non-resampled datasets for each category. The top row has plots of Reuters categories while the

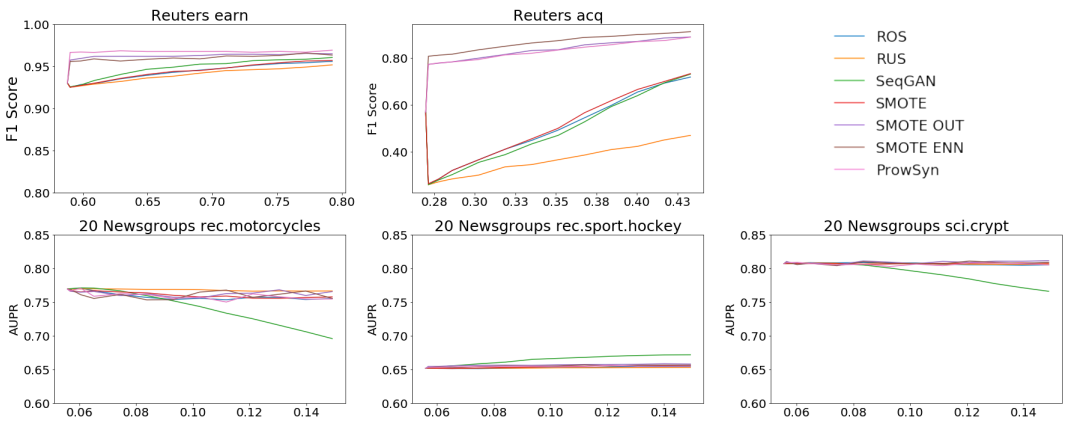


Figure 7.2. Evaluation Results of Different Resampling Methods on Reuters and 20 Newsgroups (x axis is the ratio of minority to majority samples)

bottom row is for 20 Newsgroups categories. In Reuters, for both earn and acq, we see that SeqGAN performs better than ROS and RUS when comparing text-based resampling methods. However, SMOTE-based methods are at par with SeqGAN in earn. In acq, the SMOTE-based methods are superior to text-based resampling methods.

In Figure 7.2, for the Newsgroups experiments, we observe that most resampling methods perform similarly. Note that the range of AUPR on the y-axis for the first row are from 0.60-0.85. The text-based resampling methods are ROS (Random Oversampling), RUS (Random Undersampling), and SeqGAN. In rec.motorcycles, all resampling methods perform worse than the baseline. In this case, SeqGAN's performance declines faster than the other methods. In rec.sport.hockey, all the resampling methods perform similar to the baseline of no resampling. SeqGAN outperforms all the other methods. In sci.crypt, SMOTE-based resampling methods perform better than the baseline. SeqGAN's performance drops as the amount of resampling increases. This is likely due to poorer quality of texts generated by SeqGAN. Even among the five datasets, we do not see one method consistently outperforming the others and it is difficult to predict beforehand which resampling method is best suited

to a particular classification task.

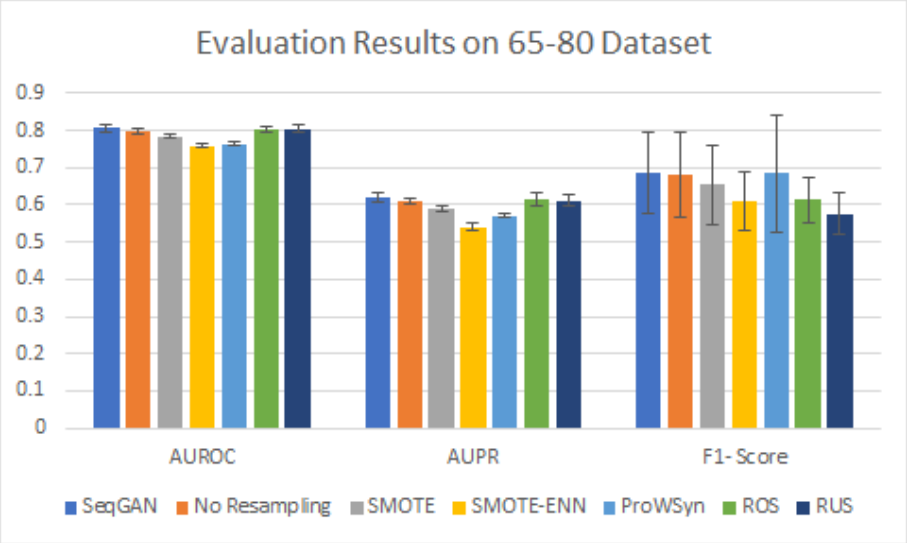


Figure 7.3. Evaluation Results of Different Resampling Methods 65-80 Dataset

We compare the resampling methods in our gender prediction task. Specifically, we report on two sets of experiments to compare the performance of SMOTE-based and SeqGAN text-based methods for resampling. We use 5-Fold cross-validation for evaluation with AUROC, AUPR, and F1-Score as the performance metrics. The 65-80 age group has an imbalance of approximately 25% male.

7.5.3 Experiment 1: SMOTE-based Resampling

We evaluate the capability of some SMOTE-based resampling techniques. SMOTE Edited Nearest Neighbor Rule (SMOTE-ENN) handles class imbalance by removing samples from both the majority and minority class [28]. SMOTE-Out considers the

nearest majority and minority example to create synthetic data [131], and ProWSyn generates weights for the minority samples based on boundary distance [24]. We generated synthetic samples so as to balance the two classes.

7.5.4 Experiment 2: Text-based Resampling

From each male user profile, we sampled 20 words with high TF-IDF values to represent the individual male user as input to the SeqGAN. Using SeqGAN, we generated 500 sequences of 20 words each which is the same sequence length used in [258]. Thus, we generated 500 synthetic male profiles for each fold of the 5-fold cross-validation. We used the implementation of SeqGAN with a CNN in the discriminator network and an LSTM in the generator network.[4]

We utilized XGBoost (with parameters set to a learning rate of 1, estimators of 9, a max depth of 5, subsample of 0.99, min-child-weight of 5, scale-pos-weight of 3, seed of 3, and gamma of 3) [54] after testing multiple configurations. We used XGBoost instead of a neural network such as DNN for this problem because XGBoost performed well and is a powerful ML algorithm also used in many papers and competitions. In the health domain, interpretability of models is vital to their practical usage and so we use XGBoost instead of neural networks which are not as easily interpreteble. Though there has been recent work on explainable neural networks, that is beyond the scope of this chapter. We compare SeqGAN against baselines with no resampling, resampling with SMOTE, SMOTE-Out, ProWSyn, and SMOTE-ENN as shown in Figure 7.3. WE did not find significant differences when parameters were varied for the SMOTE-based baselines. In Figure 7.3, we see that SeqGAN does not suffer from the sub-class problem and outperforms SMOTE and SMOTE-ENN in terms of AUROC, AUPR, and F1-Score. Text-based resampling methods of ROS and RUS perform very similarly to SeqGAN. XGBoost without resampling is second

[4] `https://github.com/bhushan23/Transformer-SeqGAN-PyTorch/blob/master/seq_gan/`

only to the text-based resampling methods. However, we expected this as XGBoost has a parameter known as 'scale-pos-weight' that varies the ratio of positive and negative examples. This allows the algorithm to better control for imbalance than many classic supervised learning algorithms.

7.5.5 Using Both Resampling Techniques

We find that combining both actually worsens the performance compared to only using SeqGAN. This is not surprising, as using resampling methods on the topic representation had a similar effect on the unaugmented text data.

7.6 Contributions

In this chapter, we leveraged content data for gender prediction. We represented users through user profiles and then topic profiles. We also tackled the imbalance problem of underrepresented male users by using a content-based technique. Thus, we see that content is effective not only to represent users and predict their characteristics, but also for improving the imbalanced classification problem.

We utilized the discrete sequence generation capabilities of SeqGANs to develop synthetic samples of the male minority class that would effectively represent the similar interests of other male users in the dataset. The experimental results showed that SeqGAN outperforms SMOTE-based resampling techniques when combined with the predictive power of XGBoost. In the future, we will explore other resampling techniques through the use of GANs, better text-summarization strategies to reduce the length of the input to SeqGAN, and more efficient methods of using higher sequence lengths with SeqGAN.

Experiments with user profiles can easily contain more than thousands of words per user instance which makes it infeasible to continuously capture a good representation of the user's full word set using the current architecture. The issue of the

scalability of SeqGAN for generating larger text samples is a limitation for its use in practical applications such as resampling.

CHAPTER 8

OVERCOMING DATA SPARSITY IN PREDICTING USER CHARACTERISTICS FROM BEHAVIOR THROUGH GRAPH EMBEDDINGS

8.1 Overview

As mentioned previously, understanding user characteristics such as demographic information is useful for the personalization of online content promoted to users. However, it is difficult to obtain such data for each user visiting the website. Since demographic data for some users can be collected, their behavior can be used to predict the attributes of unknown users. In this chapter, we focus on behavior-based approaches for representing users while also reporting on some combined techniques. We also identify features that are able to represent minority class users well.

In the domain of online news consumption, we can infer the attributes of users from the URLs of the articles they view. Most of the existing models take a supervised learning approach to this modeling task. However, by representing the user-URL interactions with a network, we can convert it to a semi-supervised learning problem and represent users through embeddings. Graph embeddings have become very popular in recent years, with research mainly focusing on algorithmic developments. However, while we have an intuitive understanding of the problems they may overcome, such as data sparsity, this problem remains unexplored in the domain of demographic prediction using behavior in online news consumption. In this chapter, we first investigate the effectiveness of using user embeddings generated from network representation learning for prediction by first comparing its performance with other

traditional feature sets, including content and item-based features. We find that the embeddings can represent a user generally by performing two prediction tasks, (1) gender prediction (classification) and (2) age prediction (regression). Second, we explore the advantages of using these embeddings over the other methods in two cases of data sparsity, where (1) the training and testing sets of users are temporally split and (2) the user labels are imbalanced. In both these cases, we show that the embeddings outperform the baseline. Finally, we further demonstrate these points by reporting on a study in which we predict future subscribers based on user behavior and past subscription information. In this problem, the dataset is both temporally split and imbalanced, and we see improved performance in the case of embeddings as opposed to the traditional features. This chapter has been submitted as a paper to a conference and is currently under review.

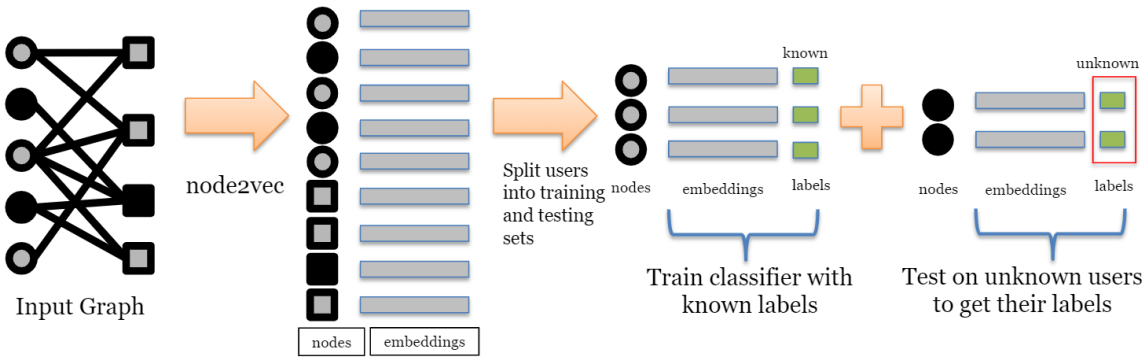


Figure 8.1: Pipeline. The circular nodes represent users and the squares are URLs. The solid black nodes are unlabeled, whereas the gray nodes are labeled.

8.2 Introduction

To improve the user experience, content-hosting organizations strive to personalize the content delivered to users in their news feeds, email newsletters, and advertisements. One such example of personalization is behaviorally targeted advertising [204, 118]. It has been stated that behavioral targeting gains double the traffic than their simpler counterparts [118]. However, for such personalization, user demographic information such as age and gender is often useful [204]. Companies can collect demographic data from some of the users, but, for the majority of content consumers on these sites, demographic information is difficult to obtain [204, 118, 75]. In response, demographic prediction techniques have been proposed in several studies [204, 118, 176, 126, 75], with most of these works focusing on feature engineering. The reason for this effort towards exploring features is that users are not directly represented in their behavior. While we can get a list of clicks on webpages from users, and metadata associated with those webpages, such as content, category, and topic, aggregating them to represent a user is not trivial and can be done in many ways. Thus, for demographic data prediction, multiple methods have been proposed for user representation. While click-level data can be used to predict users' demographics, we focus on item-level predictions as the insights are more useful to the content creators and curators. We also find that item-level features i.e., URLs are more indicative of users' gender compared to low-level content-based features such as bag-of-words (BoW), which is similar to the findings in [204]. With the advent of online content consumption, there arises an opportunity to infer the demographic information of these unknown users based on their consumption behavior. In such a scenario where the demographic data of some users is available, predictive modeling of unknown users' demographic information is possible.

One of the ways that content-providing companies can earn revenue from online services is using paywalls or a subscription service [184, 48, 99]. In the case of a

paywall, some or all content is locked behind a subscription service, and a user can only access it after paying a subscription fee. Thus, it can be useful for revenue purposes to predict whether a user will subscribe based on their behavior. We, therefore, predict if a user will subscribe to the online service based on their behavior. This task is particularly challenging due to the data sparsity issues, as defined in this chapter.

Network representation learning has become very popular for classification, recommendation, and link prediction problems recently with deployments even at the industry scale [76]. However, their application towards demographic modeling is underexplored. The methods used in the literature either use different feature sets [75, 204] or bipartite graphs [118] with gender predicted through a Bayesian framework. In this chapter, we provide deeper insights into how embeddings leaned from a bipartite graph perform in cases of data sparsity in the domain of online news consumption. Given the attributes of some users, we predict the attributes of unknown users by building a network of all the users, including known and unknown. We learn the embeddings of users and URLs using a network representation learning technique, then fit a regressor on the known users' attributes. The unknown users' attributes is predicted using the model built in the training step, as shown in Figure 8.1. In this chapter, we will explore and demonstrate the effectiveness of representing users with embeddings learned from network representation with the help of two research questions.

RQ1: *How do embeddings perform in comparison to other feature sets?*

First, we will show that user embeddings are at par or better than other methods of representing users, including item-level and content-based representation.

RQ2: *Do the user embeddings suffer from the effects of data sparsity?*

Through this research question, we argue that when the data is sparse (temporal split and imbalance), the user embeddings outperform other representations.

RQ3: *How do the embeddings perform when both types of data sparsity are combined?*

Finally, we demonstrate the effectiveness of embeddings by predicting subscribers, in which the data is both temporally split and imbalanced.

8.3 Related Work

With the recent advancements in representation learning on graphs [260], an opportunity to use the network of users and URLs for user representation arises. Specifically, some of the random walk methods that are generated for the user-item graph use a tripartite graph, including aspects [109], a correlation graph to build a biased version of the PageRank algorithm [98], nearly uncoupled Markov chains to build a personalized recommendation for users [188], and produce embeddings for bipartite networks [92]. Deepwalk [202] was one of the earliest embedding techniques proposed, and it was based on word2vec [172] and generated sentences using random walks. Node2vec [103] is similar to deepwalk, however it customizes breadth-first and depth-first sampling during the random walk process. Both deepwalk and node2vec use a skipgram model for learning embeddings. Metapath2vec [73] uses a metapath-based random walk with heterogeneous features along with a hetegeneous skipgram model for generating embeddings. On the other hand, LINE [229] generates embeddings by preserving the first and second order proximities between pairs of nodes. Among the methods used for unsupervised representation [260], we use a random walk based method node2vec [103]. While we compare the features in RQ1 against node2vec, deepwalk, and metapath2vec, we only use node2vec for RQ2 instead of the better performing methods due to space constraints. Node2vec represents random-walk based embedding methods well and does not use any additional information about the nodes such as node type, content, and item metadata used in some of the state-of-the-art methods.

One of the data sparsity problems we consider is of imbalance, in which fewer samples of minority users exist in the dataset. A way of dealing with the imbalance problem is resampling the data, i.e., changing the distribution of the training data such that the classifier can better learn the minority class. Various resampling techniques, including Random Oversampling, Random Undersampling, and Synthetic Minority Oversampling Technique (SMOTE) [51] exist to improve predictive performance on data with imbalanced class distributions [180]. These methods are generally used to help a supervised classification problem. However, the idea of using semi-supervised learning for imbalanced classification has been explored in some other works. For example, Li et. al. [149] uses a transductive semi-supervised learning method for their imbalance problem. With the help of label propagation, they add a few of the unlabeled nodes to the minority class to provide more minority class samples for training. They compare their method with other transductive graph-based semi-supervised learning algorithms. Other work on this topic exists in the domain of gene function prediction as well [84, 87, 86], such as proposing a cost-sensitive neural network [85]. Haixiang et. al. [107] describe how active learning can be used for the imbalanced classification problem in a semi-supervised setting. However, in the domain of behavior modeling, the imbalance problem has not been explored.

TABLE 8.1

10-FOLD CROSS-VALIDATION WITH DIFFERENT FEATURE SETS AND METRICS

Features	Type	Precision	Recall	F1 Score	Accuracy	AUROC	AUPR
Node2vec	**UE**	**0.661 (0.0228)**	**0.660 (0.0222)**	**0.660 (0.0223)**	**0.661 (0.0226)**	**0.718 (0.0258)**	**0.710 (0.0330)**
Deepwalk	**UE**	**0.671 (0.0227)**	**0.669 (0.0222)**	**0.669 (0.0223)**	**0.670 (0.0222)**	**0.727 (0.0260)**	**0.715 (0.0351)**
Metapath2vec	**UE**	**0.666 (0.0521)**	**0.662 (0.0485)**	**0.662 (0.0498)**	**0.665 (0.0522)**	**0.720 (0.0621)**	**0.706 (0.0768)**
Title Words	CB	0.655 (0.0046)	0.650 (0.0045)	0.649 (0.0047)	0.652 (0.0049)	0.698 (0.0034)	0.679 (0.0095)
LDA 150	CB	0.606 (0.0035)	0.605 (0.0035)	0.604 (0.0036)	0.605 (0.0036)	0.637 (0.0050)	0.613 (0.0106)
NMF 1500	CB	0.621 (0.0031)	0.619 (0.0031)	0.618 (0.0032)	0.620 (0.0034)	0.660 (0.0028)	0.640 (0.0080)
Node2vec + Title Words	HG	0.667 (0.0217)	0.665 (0.0216)	0.665 (0.0219)	0.667 (0.0218)	0.724 (0.025)	0.714 (0.0324)
Node2vec + LDA 150	HG	0.663 (0.0220)	0.662 (0.0215)	0.662 (0.0216)	0.663 (0.0218)	0.720 (0.0250)	0.711 (0.0325)
Top URLs + NMF 1500	HG	0.666 (0.0044)	0.666 (0.0044)	0.666 (0.0044)	0.666 (0.0045)	0.722 (0.0030)	0.710 (0.0061)
Top URLs	**IL**	**0.668 (0.0097)**	**0.669 (0.0050)**	**0.669 (0.0048)**	**0.669 (0.0038)**	**0.723 (0.0030)**	**0.710 (0.0059)**

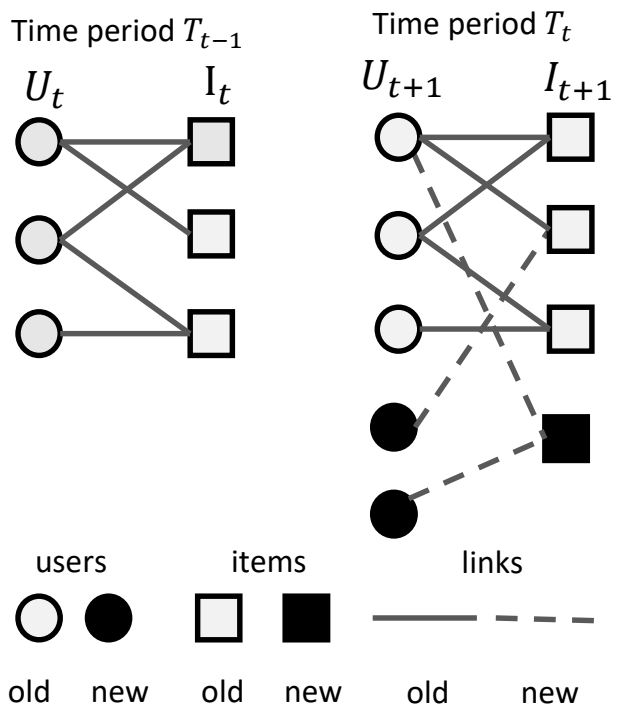


Figure 8.2. Temporal Split

8.4 Data Description

8.4.1 Dataset

We use the clickstream data of page views on an online magazine for our experiments. Most of the users in the dataset are unlabeled, i.e., their age and gender were unavailable. So, for our analysis, we have only retained the clicks made by users whose demographic data were available. In all the experiments, we kept users who had at least 10 pageviews after experimenting with different thresholds for its consistent performance with lower standard deviation. For the experiments in RQ1,

we use data from May 2018. We use the gender of 84,380 users in which 49.35% are female and the age of 64,102 users with a mean age of 58.73 and a standard deviation of 15.97 for the age experiments. Thus, while the distribution of gender is quite balanced, there are more users in the range of 60-80 years. For RQ2, in the week-wise split experiments and imbalance experiments, we use the data from May. In the month-long split experiment, we use the click data from May 2018 for training and June 2018 for testing.

8.4.2 Features

The features in Table 8.1 are listed under each subsection for reference.

8.4.2.1 Item-Level Features (IL)

We can represent users as a feature vector of URL items with the help of an indicator function. The user representation matrix U is given with the indicator matrix I, in which the URLs that the user views are indicated by counts. Let I have the dimensions $n * m$, where n is the number of users and m is the total number of URLs that are present in the click dataset.

— **Top URLs**: set of most popular URLs based on the number of total clicks on them

8.4.2.2 User Embeddings (UE)

We generate user embeddings by training a bipartite graph of users-items using node2vec. For our experiments, we use the same hyperparameter setting throughout with 100 walks per source of length 20 each, and a window size of 10. The p and q parameters of node2vec are set to the default of 1. In fact, similar to some of the reports in the study by [47], we found that the different values of p and q do not affect the performance significantly. The embeddings generated have 128 dimensions. We

also use Principal Component Analysis (PCA) [7] in RQ2, which is a dimensionality reduction method using Singular Valued Decomposition on centered data. Here also, we represent users by reducing the user-URL adjacency matrix to 128 components for each user in all the experiments. We compare PCA with node2vec as both of them are matrix factorization techniques [208] and both use clicks from both the training and testing for generating embeddings sets unlike the traditional features.

— **Node2vec Embeddings**: Embeddings generated from the user-URL bipartite graph

— **Metapath2vec [73] Embeddings**: Embeddings generated from the user-URL bipartite graph using the metapath user-item-user

— **Deepwalk [202] Embeddings**: Embeddings generated using from the user-URL bipartite graph

— **PCA**: Principal Component Analysis (with 128 components, same as the size of the embedding vector generated through node2vec)

8.4.2.3 Content-based Features (CB)

We represent the URLs themselves in terms of their content as matrix C of dimensions $m * r$, where r is the dimension of the URL-based features (e.g., r words or r topics). In this case, the user matrix U is represented by $U = I.C$ and has the dimensions $n * r$. Thus, the user representation is calculated by averaging the item representation of all the items that are associated with the user. This becomes the user feature vector U and is the input to the classification model.

We explore the bag-of-words features generated from the title of the articles instead of the body for a smaller dimensionality. As [204] have noted, these features do not work as well as the others on their own (from Table 8.1). They have, however, found that representing users as a vector of topic probabilities derived from the articles they read did better than the bag-of-words model. Therefore, we also train an

LDA (Latent Dirichlet Allocation) [36] and NMF (Non-negative Matrix Factorization) [15] model to derive the topic probabilities of documents and represent users by averaging the probabilities of the articles that each user reads. After experimenting with different topic numbers, we show the results in Table 8.1.

— **Title Words**: Bag-of-Words representation of articles from titles

— **LDA 150**: LDA topic model with 150 topics

— **NMF 1500**: NMF topic model with 1500 topics

8.4.3 Heterogeneous Features (HG)

We can also incorporate other features into the feature vector by concatenating (augmenting) them to the user matrix U. For example, if matrix L of size $n * l$ denotes the user-location matrix with l total locations, the user representation would be given as $[U|L]$ and have dimensions $n * (m + l)$. Other features based on time and browser can be used to augment this matrix, in order to incorporate more of the heterogeneous features. In Table 8.1, the heterogeneous features have a + in the Features column and are generated through concatenation.

— **Node2vec Embeddings + Title Words**: Concatenate user embedding with content-based feature

— **Node2vec Embeddings + LDA 150**: Concatenate user embedding with content-based feature

— **Top URLs + NMF 1500 Topics**: Concatenate item-based with content-based feature

8.5 Model Description

In this section, we describe the prediction framework for the experiments and the pipeline shown in Figure 8.1.

8.5.1 Base Model

We use Logistic Regression for gender prediction as it is one of the best performing classifiers for our problem. For age, we use Beta Regression since age lies between 0 and 100 for our users. This is a better fit than linear regression, which does not respect the bounds of age and could potentially predict a negative or unrealistically high age. All the user representations, including the title words, topic features, embedding features, and heterogeneous features, are trained on either of these models for gender and age prediction. We measure performance on gender prediction mainly with accuracy, but also report on precision, recall, f1 score, area under receiver operator characteristics (AUROC) and area under precision-recall (AUPR). For age prediction, since it is a regression problem, we report on the metrics of mean squared error (mse), r^2, mean absolute error (mae) and root mean squared error (rmse). For cross-validated experiments, the mean of the metrics is reported with the standard deviation given in parentheses.

8.5.2 User Embeddings

We generate a bipartite graph $G(V, E)$ shown in Figure 8.1 where V are the user and URL vertices, and the edge E between a user and a URL exists if the user viewed that URL. This edge can be weighted by the number of clicks the user made on that URL, but in our experiments, we do not use weighted edges since the weighting did not improve the performance. This is likely due to the number of click occurrences by a single user appearing in the click log not being reliable due to pages being reloaded by the browser automatically if a user has the tab open (e.g., when the computer restarts). The users in the training set have labels assigned to their corresponding vertices, whereas the users in the testing set are unlabeled. We generate node embeddings by the following steps as outlined by node2vec [103]:

1. Random walk on the graph starting at each vertex of the graph with a fixed walking length (20) and number of walks per starting vertex (100).
2. Input these random walks into the word2vec [173] model with a given context size (10) to generate embeddings for each node.
3. Train a classifier with the embeddings of the training vertices as the input feature vector and predict gender based on the embeddings of the testing vertices.

While embeddings are generated for URL nodes as well, we do not use them in our model. Once the embeddings are generated, they can be used in the baseline predictors for training and testing, as shown in Figure 8.1.

8.6 Analysis

8.6.1 Comparison of Different Feature Sets

In this section, we answer RQ1: *How do embeddings perform in comparison to other feature sets?*

For this question, we focus mainly on gender prediction because it is a larger dataset as well as for space considerations. Table 8.1 shows the performance of different feature sets for gender prediction. Through experimentation, we discovered that only using the top 8% most popular URLs to represent users in the Top URLs representation gives the best performance and slightly drops if we include more URLs. This is likely due to logistic regression overfitting on a highly dimensional feature vector. Table 8.1 is not an exhaustive list of all possible feature sets but lists some of the best-performing ones. Overall in both age and gender prediction, the performance of embedding methods including deepwalk, node2vec, and metapath2vec are at par with Top URLs, the best performing features among the traditional features. Unlike the top 8% of the URLs for gender prediction, the best performing threshold for age prediction is only 6% of the most viewed URLs. We observe that the AUPR values in Table 8.1 are high because of the precision and recall being balanced through all the

feature sets. We also see that node2vec features get a slight boost when combined with content features, suggesting that models incorporating heterogeneous features have the potential to outperform models that only use one type of feature.

TABLE 8.2

AGE FEATURES

Metrics	Node2vec	Top URLs	Metapath2vec	Deepwalk
mse	201.6 (9.4)	205.6 (5.8)	198.6 (2.4)	197.5 (3.3)
r2	0.208 (0.04)	0.197 (0.02)	0.221 (0.01)	0.226 (0.01)
mae	11.2 (0.3)	11.3 (0.2)	11.1 (0.1)	11.0 (0.1)
rmse	14.1 (0.3)	14.3 (0.2)	14.1 (0.1)	14.1 (0.1)

8.6.2 Data Sparsity Conditions

In this section, we answer RQ2: *Do the user embeddings suffer from the effects of data sparsity?*

In the domain of user modeling, many times, the data available is sparse, either because users do not visit many URL items or because fewer users of a particular demographic group engage with the website. These problems impact the performance of models on certain predictive tasks. The following subsections describe two examples of tasks impacted by data sparsity and showcase how traditional supervised learning frameworks succumb to these issues, while the user embeddings in our framework

are resilient to them. We focus our experiments on Top URLs and node2vec representations since Top URLs are the best performing of the traditional features and node2vec has the lowest performance among the embedding methods. The rationale here is that the features that do not perform as well during cross-validation will likely perform even worse during data sparsity situations. By excluding combined features, we can focus our analysis on the comparison of performance of embeddings generated using user-item clicks versus feature vectors generated by the items, without having to control for the effect of other features such as content.

8.6.2.1 Temporally Split Training and Testing Set Users

Given some users whose demographic information we do have, we can infer the demographic information about other users that access the same set of articles, as discussed previously. However, since new articles are released weekly, the time duration in which a URL is actively viewed is limited. When this time period is less than the time window in which we train and test user features for prediction, item-level features are no longer feasible for use due to very few users in the future actually viewing them. Thus, the sparsity issue of user-URL clicks, in which the number of users viewing x URLs follows a power-law distribution, is exacerbated when a temporal split is considered. This problem is motivated by the observation that days far away from each other have a smaller intersection of URL clicks than closer days as seen in [105, 154].

Let the users at time t be represented by U_t and the URLs at time t be represent by I_t as shown in Figure 8.2. First, we compare two of the best feature sets on the cross-validated experiments, namely Top URLs and node2vec embeddings, which use structural rather than content-based features.

TABLE 8.3

10-FOLD CROSS-VALIDATION WEEK-WISE FOR GENDER (ACCURACY)

Week	Top URLs	Node2vec	PCA
1	0.596 (0.008)	0.589 (0.019)	0.589 (0.007)
2	0.604 (0.006)	0.597 (0.014)	0.601 (0.005)
3	0.648 (0.004)	0.645 (0.019)	0.647 (0.003)
4	0.620 (0.005)	0.618 (0.014)	0.606 (0.006)

TABLE 8.4

10-FOLD CROSS-VALIDATION WEEK-WISE FOR AGE (MSE)

Week	Top URLs	Node2vec	PCA
1	231.1 (7.42)	228.3 (9.19)	227.4 (4.45)
2	227.2 (7.56)	222.4 (6.68)	225.9 (4.59)
3	223.6 (4.26)	219.5 (6.16)	221.3 (4.71)
4	230.4 (5.37)	224.9 (4.297)	224.4 (5.38)

TABLE 8.5

WEEK-WISE SPLIT FOR AGE

Train on	Test on	Top URLs (6%)				Node2vec				PCA				N2v_strict			
		mse	r^2	mae	rmse	mse	r^2	mae	rmse	mse	r^2	mae	rmse	mse	r^2	mae	rmse
Week 1	Week 2	279.4	-0.0890	14.0	16.7	225.9	0.1023	12.0	15.0	273.4	-0.0860	13.4	16.5	240.4	0.0745	12.5	15.5
Week 1	Week 2+3	278.7	-0.0656	14.0	16.7	220.4	0.1128	11.8	14.9	252.2	-0.0153	12.8	15.9	245.6	0.0573	12.7	15.7
Week 2	Week 3	235.5	0.0846	12.5	15.3	221.4	0.1038	11.8	14.9	227.4	0.0807	12.2	15.1	239.0	0.0794	12.4	15.5
Week 2	Week 3+4	242.3	0.0670	12.6	15.6	217.7	0.1301	11.7	14.8	225.9	0.0973	12.1	15.0	239.0	0.0890	12.4	15.5
Week 3	Week 4	279.3	0.0185	13.5	16.7	234.2	0.1115	12.1	15.3	245.6	0.0708	12.6	15.6	246.1	0.1029	12.4	15.7
May	June	256.9	0.0206	13.2	16.0	221.1	0.1164	12.1	14.9	231.7	0.0744	12.5	15.2	242.1	0.0757	12.6	15.6

TABLE 8.6

WEEK-WISE SINGLE-SPLIT FOR GENDER (ACCURACY)

Train on	Test on	Top URLs	Node2Vec	PCA	N2v_strict
Week 1	Week 2	0.553	0.629	0.554	0.557
Week 1	Week 2+3	0.553	0.613	0.603	0.552
Week 2	Week 3	0.602	0.623	0.628	0.578
Week 2	Week 3+4	0.580	0.630	0.628	0.579
Week 3	Week 4	0.584	0.609	0.599	0.585
May	June	0.540	0.639	0.605	0.575

Tables 8.3 and 8.4 shows these results for gender and age prediction respectively. For node2vec and PCA embeddings are generated using the behaviors of the individual week only. We see that all the methods are similar to each other in performance.

Then, we train on users in an earlier week and test on users in a later time period. In the case of node2vec, PCA, and top URLs, we use behavior from only the weeks in the training and testing sets. For example, in the first row of Table 8.6, clicks from both weeks 1 and 2 are used to generate node2vec, PCA, and top URLs representations. Similarly, in the second row, weeks 1, 2, and 3 were used to generate node2vec, PCA, and top URLs representations. Note that the behavior of both the training and the testing weeks are available to all the models at the time of prediction. However, in the case of top URLs, if a URL was only clicked on during the testing week, the associated column would have zero values for all the users in the training set. This is a limitation of the top URLs representation as it cannot take advantage of all the clicks for training the model. Tables 8.5 and 8.6 shows these results, where each row shows the prediction result across a single training-testing split based on

time.

We find that Top URLs drops in performance much more dramatically than node2vec when the training and testing time periods are different for both gender and age prediction problems. This is consistent with our expectation of semi-supervised learning being able to deal with sparse data better. The last column N2v_strict shows the performance of node2vec if we do not include nodes corresponding to URLs that only appear in the testing week(s). We see that the performance drops considerably in both gender and age experiments, which emphasizes to us the importance of being able to include all clicks in the model.

8.6.2.2 Imbalanced Classification

While the previous problem of temporally split training and testing suffers from a sparsity in the features, imbalanced classification problems suffer from a sparsity in the number of samples associated with the minority class. When not all user demographics are equally represented in the data, traditonal classifiers struggle to predict the minority class users accurately. However, the effect of the imbalance is not as prominent when using network representation learning. To evaluate the effectiveness of the performance of network embeddings on imbalanced data, we perform experiments by artificially creating an imbalance in the dataset. We subsample male users such that the ratio of males to all users is 0.10. To do a systematic analysis, we repeat the experiments 9 times with different subsets of male users.

In most of the cases, using SMOTE for resampling the minority class worsens the performance at different resampling levels, and so we do not resample the data for the baseline. Figure 8.3 shows the performance of the baseline classifier using Top URLs features and node2vec for each subset. For every subset except the second one, node2vec outperforms the baseline classifier. The error bars indicate the standard

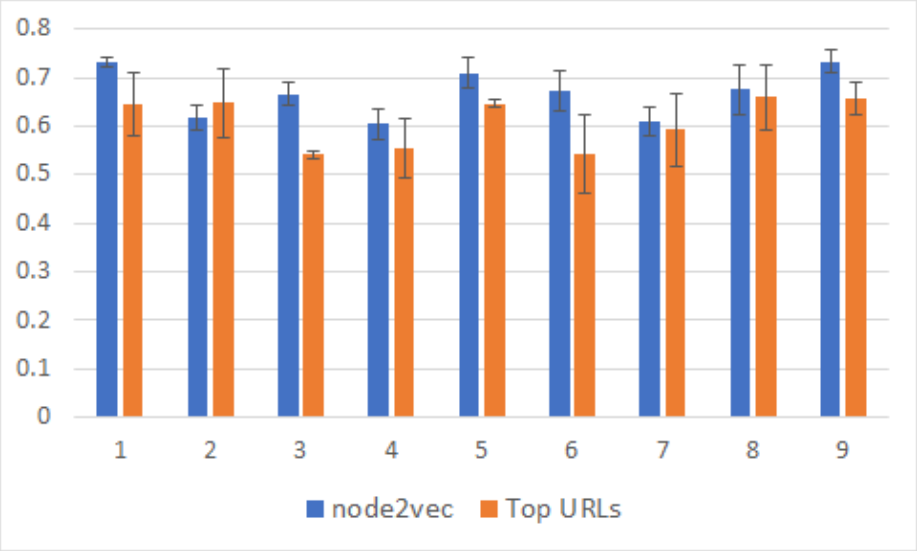


Figure 8.3. Imbalanced Classification Comparing Baseline (Top URLs) and node2vec on a 10-Fold Cross-Validation

deviation. Thus, we see that node2vec does not suffer from the problem of imbalance as much as the Top URLs features. Since the node2vec embeddings are trained on all the users, in the training as well as the testing set, presumably more data is used to generate embeddings compared to the Top URLs feature vector, which does not leverage samples from the testing set to ameliorate the sparsity issue. Instead of explicitly labeling minority nodes in the testing set and including them in the training set as is done by [149], the minority nodes from the testing set are implicitly considered through the random walking.

8.7 Predicting Subscribers

In this section, we predict subscription to answer RQ3: *How do the embeddings perform when both types of data sparsity are combined?*

Most of the previous studies on subscription have focused on whether users are willing to pay for subscription services, what kind of content they are willing to

TABLE 8.7

SUBSCRIPTION PREDICTION WITH TEMPORALLY SPLIT TRAINING AND TESTING SETS

Features	Resampling	Type	Precision	Recall	F1	AUROC	AUPR
Node2vec	None	UE	0.5088	0.5922	0.5060	0.8011	0.0160
Node2vec	SMOTE (0.02)	UE	0.5047	0.7412	0.4191	0.8042	0.0164
Top URLs	None	IL	0.4984	0.4990	0.4987	0.6401	0.0048
Top URLs	SMOTE (0.1)	IL	0.5045	0.5192	0.5056	0.6888	0.0071
Top URLs	ROS (1.0)	IL	0.5024	0.5714	0.4780	0.6632	0.0059

pay for and on what platforms, and which users are more likely to pay (various demographic variables), etc. However, studies on predicting subscriptions for online news consumption are lacking. This is a particularly challenging problem as only a very small percentage of the total users that visit a website subscribe.

A significant body of research focuses on understanding the factors that contribute to subscription. For example, one study investigates the reasons behind the content that is not free to access online [110]. Another study also explores what content is locked behind a paywall [185]. One of the studies analyzes the relationship between willingness to pay for online news and various predictors such as demographic information, media use, news interest, and traditional newspaper subscription [100]. Similarly, a study has identified that young users are more likely to pay when online services use micro-payment strategies that allow users to purchase content on an article and page basis [102]. According to another study, demographic factors such as age and gender influence the likelihood of users paying for online news services but overall found that users are more likely to pay for printed than online versions of

newspapers [59]. Another paper identifies that the format of the online content, as well as device ownership, are important for assessing the willingness of users to pay for the online newspaper service [31].

Some works go beyond the problem of prediction subscription and focus on the problem of continued subscription. For example, a paper describes a method for making recommendations to users that will extend their subscription periods [121]. Another study identifies the factors that discriminate between subscribers who continue their services and those who unsubscribe from the content [196]. An analysis of the variables that help identify users who switch their service as opposed to those who continue their subscription shows that users who switch subscriptions rely on word-of-mouth sources generally [127].

8.7.1 Problem Formulation

Given the users and their behavior in a particular month, can we predict, from the clicks of the viewers in the next month, which of them will subscribe? Let users U_t be the users with activity in the earlier month. Let users in U_{t+1} be the users with activity in the next month. Of these users, the testing set $U_{Testing}$ comprises of the users in U_{t+1} that are not in U_t. In other words, $U_{Testing} = U_{t+1}$ - U_t. For the users who subscribe in the first month denoted as S_t, we remove clicking activity after the subscription timestamp from the dataset. Thus, only the clicks that seemingly lead to subscriptions are retained. The subscribers form the minority class, whereas the non-subscribers are the majority class. Since it is not possible to differentiate between the users who never subscribe and the users who subscribe in subsequent months, we label non-subscribers in the training set as those users with no activity in the next month. That is, only the users in the set U_t - U_{t+1} - S_t are labeled as the majority class. The ground truth for the users in the testing set $U_{Testing}$ are labeled as subscribers and belong to the set S_{t+1} if they subscribe in next month, whereas

they are non-subscribers if they do not subscribe in the next month.

8.7.2 Data Description

The percentage of users that subscribe in a given month is less than 2%. We use click logs from June 2019 and July 2019 for the following experiments. Thus, this problem is temporally split and imbalanced. We keep users in each month with at least 10 pageviews in that month, similar to the gender and age prediction problem. This leads to 189,303 users in June and 134,504 users in July. June has a total of 3,259,664 pageviews, and July has 2,324,409 pageviews. The parameters of node2vec and Top URLs are the same as those used in the gender prediction experiments.

8.7.3 Experiments

For the analysis, we demonstrate the performance of both user embeddings and the Top URLs features for the task of predicting subscribers. We also include various resampling strategies that would help alleviate the imbalance problem. Table 8.7 shows the performance of these methods with the resampling method noted in a separate column. We see that the user embeddings perform much better than Top URLs without any resampling. SMOTE and Random Oversampling (ROS) are two methods for resampling minority classes. The synthetic samples are augmented into the Top URLs feature set with the sampling ratio provided in parentheses as the ratio N_{rm}/N_M, where N_{rm} is the number of samples in minority class after sampling and N_M is the number of samples in the majority class. ROS improves the recall slightly. Thus, including resampled rows in the dataset leads to the classifier making more predictions on the minority class (subscribers). However, the performance of resampled Top URLs features is still worse than that of the user embedding features. Resampling the embeddings using SMOTE does not affect the performance much.

Thus, node2vec is able to improve upon the performance of the URL features

TABLE 8.8

SUBSCRIBERS RESAMPLING 3-FOLD CROSS-VALIDATED

Resampling	AUROC Mean (Std)
None	0.7018 (0.008)
ROS 2	0.7005 (0.015)
ROS 3	0.7121 (0.002)
ROS 4	0.6940 (0.012)
ROS 5	0.7011 (0.012)
ROS 10	0.6887 (0.016)
Balanced Batch Undersampling	0.6612 (0.026)

greatly. While some resampling strategies such as balanced-batch sampling have been proposed [253], they do not improve performance in this case. This is likely because data resampling strategies revolve around balancing the ratio of nodes in the training set that appear in the random walks. However, the samples to be predicted are in the testing set, and the proportion of testing set nodes in the random walks is either the same as without resampling or diminished. Thus, designing graph embedding methods that can improve the representation of minority class nodes in this problem is part of our future pathway.

User embedding features can be resampled in two ways. [253] introduced the concept of balanced batch sampling, wherein, in each batch of training the word2vec, the majority class nodes are undersampled such that there are equal number of instances of majority and minority classes in each batch before the negative sampling step. The second resampling strategy we can use with node2vec is oversampling the minority nodes in the graph and creating synthetic nodes with edges that are identi-

cal to the nodes they are replicating. In other words, we introduce fake subscribers that replicate the behavior of randomly selected real subscribers in the training data. This leads to more instances of minority class nodes sampled in the random walks. We can decide the resampling hyperparameters to be used by doing a k-fold cross-validation on the training set. In our experiments, we have set k as 3. We find that Random Oversampling the minority class 3 times improves the performance slightly over the performance of node2vec without resampling. We also observe that balanced batch undersampling performs the poorest in this task. While assigning subscribers to the minority class in the training set is easy, the problem of assigning users from the earlier month as subscribers or non-subscribers is not trivial. A user that does not subscribe in a month may subscribe in a subsequent month. Does that imply that the user should be classified as a non-subscriber? Defining a user based on their subscription status in the current month could be one way to define the classes. However, to be safe, we actually use the users that only have activity in June, but do not subscribe in June, as the majority class (non-subscribers). We argue this is valid because users that do not have clicks in July are highly unlikely to subscribe in July. In the next month, we define subscribers (minority class) as those who subscribe in that month, whereas users that do not subscribe in July are the majority class. We also restrict the clicks of the June subscribers that will be included in the model only until the time of subscription.

8.8 Contributions

Through this chapter, we showed how content, behavior, and combined approaches for user representation are all effective for predicting users' demographics to varying degrees. We investigated different methods to represent these features such as topic modeling, bag-of-words, and graph embeddings. Behavior-based features performed better than content-based features in this dataset. Comparing URL features with

graph embeddings, we saw that embeddings were resilient to data sparsity issues of temporally split data and imbalanced classes. Thus, the use of graph embeddings is promising for the challenge of underrepresented user.

By representing the user-URL data through bipartite graphs, we were able to convert the problem from a supervised learning problem with URL features to a semi-supervised node classification problem using the graph data. A possible explanation for why the URL information is more predictive than the actual content is that many URLs are shared on social media networking or other means which have a higher demographic bias. In that case, an article could have a higher occurrence in a demographic group, irrespective of the content. We also discovered that only the most popular URLs are necessary for gender and age prediction tasks, which is good for the scalability of regression.

From RQ1, we have seen that user embeddings can represent users well. Since these embeddings were generated in an unsupervised fashion, they could theoretically be used for any downstream application. We consider the two prediction tasks – gender (classification) and age (regression), and observe that the user embeddings perform well in all of these tasks. Thus, we see that the user embeddings were general enough to fit well to different tasks.

In our experiments with node2vec, we only kept users whose genders/ages were known. However, since the generation of user embeddings is unsupervised, we could use all the data available to us instead of restricting ourselves to a smaller subset of users. This provides a richer dataset for user representation as compared to other supervised learning methods. We added 101,338 users for age prediction and had a slight improvement in performance with an mse of 199.3 (10.3), r^2 score of 0.2166 (0.0356), mae of 11.13 (0.340), and rmse of 14.11 (0.361). Similarly, for gender prediction, the accuracy is 0.6634 (0.0217). Thus, the potential for using network embeddings for user representation needs to be further explored in the scope of the

breadth of data rather than depth of data, which requires a large amount of activity for each user. There are also potential modeling designs that can be explored based on these ideas.

However, there are some shortcomings in using this method that could potentially be addressed in the future. Most of the existing network representation techniques are transductive and thus cannot be trained on streaming data. In other words, these techniques would be useful for predicting users' attributes retrospectively, but not in real-time. For network representation techniques, new users can only be represented by retraining the model on the entire data, making it expensive to train such a model for real-time prediction problems, such as predicting the gender of a user currently visiting the website. Another disadvantage is that there is an explicit training stage for the user representation, which is not required with some other techniques, such as Top URLs. So in situations where network representation would not offer an advantage, it may be more prudent to use less expensive methods of user representation. Nonetheless, more sophisticated modeling that leverages the heterogeneous data could potentially improve performance compared to simpler network representation techniques.

CHAPTER 9

IMBALANCED CLASSIFICATION USING GRAPH EMBEDDINGS

9.1 Overview

In Chapter 8, we saw that graph embedding methods were robust to performance drops caused by an imbalanced distribution of sample classes. Inspired by this observation, we explore if graph embeddings can be used to solve the problem of imbalanced classification in the traditional setting.

Imbalanced classification is an established data science problem in which the difficulty of classifying samples is exacerbated by one of the classes being poorly represented in the data. Many methods have been proposed to solve this problem and fall into different categories, such as data-level, algorithm-level, and ensemble techniques. In this chapter, we consider mapping the imbalanced learning problem to a representation learning problem from graphs. We propose a transductive classification algorithm that constructs a graph from all the vertices, including labeled and unlabeled. We then use a network representation learning technique to generate embeddings for all the samples. These embeddings are used as features for the downstream classification task. We consider two graph construction methods, including ϵ-neighborhood and k-nearest neighbors graph construction, and evaluate on a large news data as well as a dataset that is typically used for imbalanced classification. We compare these methods with data-level resampling baselines, including SMOTE-variants, and show that our framework used with the k-nearest neighbor graph construction method outperforms other resampling techniques on all the datasets used

in this chapter. This chapter is currently under review as a paper submitted to a conference.

9.2 Introduction

With daily life activities and services such as online shopping, news, and content consumption migrating to the web, many prediction tasks rely on using internet data. One such problem is the prediction of events occurring on a particular day and location, where some of the features are derived from online news articles. Since most of the days are uneventful, this prediction problem is imbalanced. This problem of imbalanced classification is quite prevalent in the real world, with applications such as cancer identification [51], medical diagnoses [81], and detection of oil spills [214], and online world, for example, fraud detection in online banking [244], sentiment analysis in social media [9], and inferring underrepresented users' characteristics from online content consumption [228]. In many of these cases, correctly identifying the rare samples of fraudulent activity, cancerous cells, and disasters is more important than identifying the majority class instances due to a higher cost associated with false negatives than false positives. However, models trained on imbalanced data are biased towards the majority class, since the models generalize for all the data and tend to improve overall accuracy [80]. Multiple strategies exist for alleviating this problem. A survey by Rout et al. [214] groups the strategies of handling class imbalance into categories of data-level, algorithm-level, ensemble, and other techniques such as feature selection, and dimensionality reduction.

Another potentially useful approach that has not been as popular as the previously mentioned methods is semi-supervised learning. Instead of oversampling by either duplicating existing data points or generating synthetic data points from the training data, the unlabeled data is leveraged to augment the training data and ideally provide new minority class samples to train the classifier. This use of semi-supervised learning

would group it with data-level strategies. One such approach is active learning, which employs some labeling of unlabeled instances. However, this is expensive and requires external labeling [89]. Another method used is label propagation [149, 261], where some instances of the unlabeled data are marked as the minority class and included in the training set of the classifier. This idea does not require an expensive external labeling process. Instead, it relies on constructing graphs with all the samples, labeled and unlabeled, as vertices, and using the neighborhood to propagate labels. The training set is augmented with these unlabeled samples selected as the minority class for training the classifier.

Graph representation learning has recently become a popular area of research, drawing novel algorithms and insights for improved representation of vertices [202, 103]. These techniques aim to generate fixed, low-dimensional embeddings for the vertices in the graphs. Similar vertices that have homophily or are in the neighborhood of each other are represented closer in the embedding space. In our experiments, we use the unsupervised word2vec inspired graph representation method, node2vec, to learn the embeddings of the samples from the constructed graph. In this way, we do not explicitly label unknown samples as the minority class. However, the unlabeled minority class samples could ideally draw the labeled minority class samples closer in this embedding space due to higher connectivity in the graph. Thus, the steps for this approach, as illustrated in Figure 9.1, are as follows:

1. Generate a graph representation of the data that includes both labeled and unlabeled samples.
2. Jointly learn embeddings of all the samples in the data using a network representation learning technique.
3. Train a classifier with the new embeddings as features of the training set and predict on the testing set, as usually done in a supervised learning problem. The difference here is that instead of using the original features, we use a new representation of the data.

We investigate the usefulness of this approach through detailed experimentation.

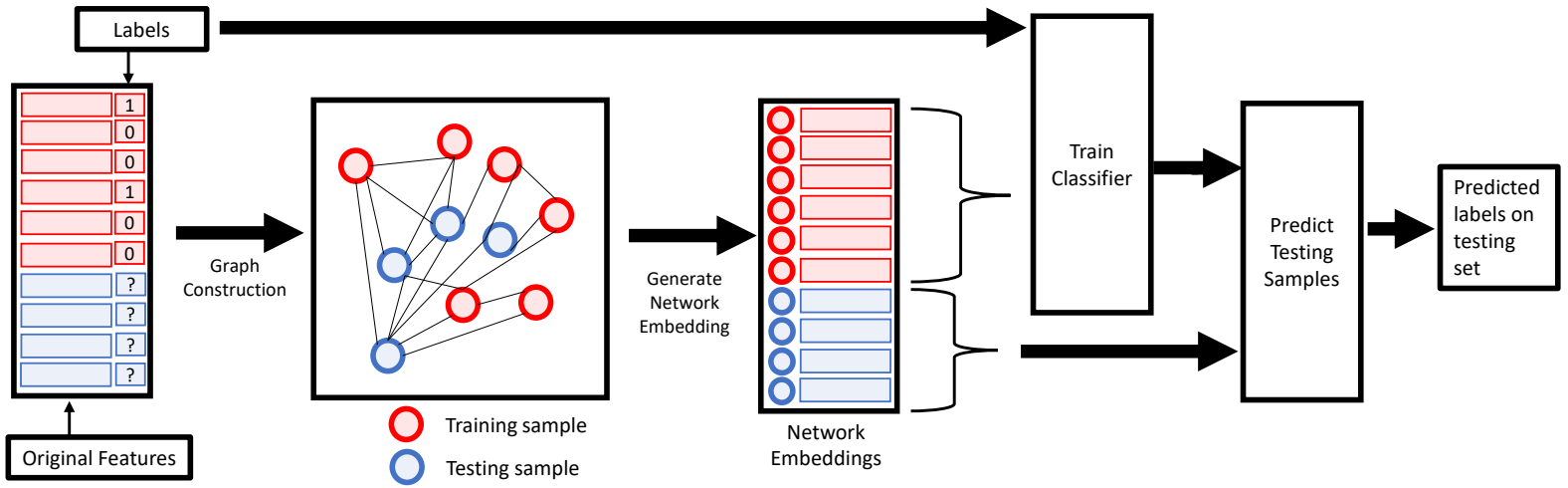


Figure 9.1: Framework

9.3 Related Work

Popular data-level strategies for reducing imbalance include oversampling, which increases the number of minority class samples in the training set, undersampling, which decreases the number of majority class samples, or a combination of both. Among the oversampling strategies, random oversampling can lead to overfitting [217]. Thus, many methods of generating synthetic minority samples have been proposed. Synthetic Minority Oversampling Technique (SMOTE) is one such technique that generates synthetic samples using the k-nearest minority samples [51]. Many modifications to SMOTE have been proposed [262, 210, 81]; however, these methods suffer from certain drawbacks. First, no strategy works better than the others on all datasets [45]. Second, the resampling ratio needs to be experimentally chosen [45]. If the resampling strategy is not suited to the dataset, it can worsen performance [45]. Another drawback of synthetically generated minority samples is that if these synthetic samples spread into majority class's decision boundary, the performance drops due to noisy samples in the training set [217].

While data sampling strategies can improve performance without modifying the classification algorithm, strategies that focus on modifying the algorithm to account for the imbalance in the data distribution are also popular. Cost-sensitive learning considers misclassification cost, generally by assigning a cost matrix to the learner [157, 144]. Another common method is modifying the algorithm to incorporate a class weight into its predictions [124]. Recently, several studies have been published that deal with class imbalance in deep neural networks, specifically [124]. Other examples of algorithm-level strategies include focal loss [156], which reshapes the cross-entropy such that easily classified samples have lower impact on the loss, and Focused Anchor Loss [21], which uses a two step loss function to combine discriminative feature learning with cost-sensitive learning.

Recently ensemble learning based solutions have become popular [134]; however,

feature selection strategies are under-explored, as noted by Leevy et al. [144]. These approaches generally optimize for imbalanced classification by using techniques such as filters, wrappers, correlation between features, information gain, and odds ratio to choose a subset of features that would better discriminate the two classes [144].

Most of the previously described methods fall in the paradigm of inductive learning. In the case of imbalance, transductive learning may lead to improved performance. Li et al. [149] use a transductive semi-supervised learning method for their imbalance problem. They do label propagation and add some of the unlabeled nodes to the minority class. They compare their method with other transductive graph-based SSL algorithms. Li et al. [150] propose a new method called Label Matrix Normalization, which uses a normalized label matrix to handle the imbalanced problem [150]. Some work on this topic exists in the domain of gene function prediction [84, 87, 86]. Frasca et al. [85] propose a cost-sensitive neural network, while Haixiang et al. [107] describe how active learning can be used for the imbalanced classification problem in a semi-supervised setting. While methods that use graph-based semi-supervised learning for imbalanced classification exist [122, 37], they do not leverage network representation techniques to generate new embeddings for the samples and address the problem of class imbalance.

Network representation learning has recently become popular for learning embeddings of graph vertices. Many different approaches exist for learning these embeddings, as described by Cai et. al [46]. One group of techniques uses random walks for generating embeddings of vertices. These methods were inspired by word2vec [173], which uses SkipGram or Continuous Bag of Words (CBoW) for the embedding generation process. Random walking methods are used to sample paths (or sentences in word2vec), providing positive pairs to SkipGram, which then embeds the vertices. The details of this algorithm are explained in Section 9.5. Deepwalk [202], node2vec [103], and metapath2vec [73] are some examples of the different methods

proposed in this family of embedding techniques. Most of these methods are transductive, which means that the entire graph, including labeled and unlabeled vertices, is required for the generation of embeddings. A new graph would have to be constructed to generate embeddings for new vertices, and the embeddings would have to be retrained. In this chapter, we use node2vec [103] for generating the embeddings after graph construction.

Semi-structured data is represented by rows and columns, where the rows are individual samples in the dataset, and the columns are each dimension of the feature vector representing the samples. We can convert this representation to an affinity matrix using a distance measure. Each entry in the matrix shows the similarity between the corresponding rows and columns. However, using the original matrix leads to a dense graph with many edges. Graph sparsification is a process in which most of the entries in the adjacency matrix are driven to zero. One of the methods to do this is the neighborhood approach. In this chapter, we use ϵ-neighborhood and k nearest neighbors (knn), a couple of popularly used techniques in the domain of graph-based semi-supervised learning [161]. We describe these methods in detail in Section 9.5. Other more recent graph-construction methods such as b-matching [122] exist; however, they have greater time or space complexity.

9.4 Problem Definition

Given a dataset $D = \{s_0, s_1, ..., s_n\}$, where each $s_i = (d_i, y_i)$ such that d_i is a feature vector and y_i its corresponding label, assume that the samples $s_i \in D$ are *iid* (independent and identically distributed). Let instances from $0...k$ be part of the training set, and instances from $k+1...n$ be part of the testing set, without loss of generality. We wish to improve the representation of these samples d_i by generating a new representation e_i for each of these samples, such that the minority class samples are more separable from the majority class. Ideally, we want $P(y_j = m|e_j) \geq P(y_j =$

$m|d_j)$ for each sample s_j belonging to the minority class m.

9.5 Model Description

In this section, we describe the steps in the algorithm for generating the embeddings, that is, a new feature representation for the samples in the dataset.

9.5.1 Graph Construction

Before graph construction, all the feature vectors are scaled between 0 and 1 using min-max normalization. This is to ensure that no single dimension biases the pairwise distances calculated between the samples. The new value $d'_{ij} = \frac{d_{ij} - min(d_j)}{max(d_j) - min(d_j)}$, where $d_i j$ is the j^{th} dimension of feature vector d_i, and d_j represents a column vector of the j_{th} dimension of all feature vectors in D.

In the first step, we construct a graph $G(V, E)$ where each sample d_i corresponds to vertex v_i. Therefore, graph G has $|D| = n$ vertices. We describe two different methods for constructing the edges E in graph G.

9.5.1.1 ϵ-Neighborhood

In this method, we compute the pairwise distance of all feature vectors, denoted as $s(d_i, d_j)$, for every $i \leq n$ and $j \leq n$. Also, $s(d_i, d_i) = 0$. These distances are used to compute the affinity matrix as follows. A threshold ϵ is chosen for the generation of edges. All pairs with distance greater than epsilon have no edges drawn between them.

$$w(i,j) = \begin{cases} 1, & \text{if } dist(d_i, d_j) \leq \epsilon \\ 0, & \text{otherwise} \end{cases} \tag{9.1}$$

Equation 9.1 is used for constructing the affinity matrix. In our experiments, we compare the following distance functions as defined in Table 9.1. We consider distance

TABLE 9.1

DISTANCE MEASURES USED TO LINK VERTICES IN GRAPHS. $x = \{x_0, x_1, ..., x_n\}$ AND $y = \{y_0, y_1, ..., y_n\}$ ARE TWO FEATURE VECTORS REPRESENTED AS VERTICES IN THE GRAPH.

Distance Measure	Formula
Manhattan	$dist_1(x, y) = \sum_i(\|x_i - y_i\|)$
Euclidean	$dist_2(x, y) = \sum_i(\sqrt{x_i^2 - y_i^2})$
Chevyshev	$dist_\infty(x, y) = \max_i(\|x_i - y_i\|)$
Cosine	$dist_{cos}(x, y) = 1 - \frac{\sum_i x_i y_i}{\sqrt{\sum_i x_i^2}\sqrt{\sum_i y_i^2}}$

functions based on different norms in the L^p space. We also include cosine distance, which is based on the definition of cosine similarity.

9.5.1.2 K-Nearest Neighbors (knn)

In this method, similar to the method above, we also compute pairwise distances between all the vertices. However, for each vertex, edges are only drawn between the current vertex and k neighboring vertices with the highest similarity. This method is more popular than the ϵ-neighborhood approach, which results in a more disconnected graph and is sensitive to the chosen value of ϵ [53].

We experimentally compare the ϵ-neighborhood and knn approaches in Section 9.7 as well as provide some insights into the graphs being constructed through measures like the number of connected components in the graph and degree distribution.

9.5.2 Generating Embeddings

The embedding generation process is based on random walks and uses a Skip-Gram model. We use the algorithm introduced as node2vec, with p and q set to 1 [103], which is the same as DeepWalk [202] but with negative sampling instead of hierarchical softmax. This is to ensure that we are doing only a first-order Markov chain. More sophisticated walks, including higher-order Markov chains, can be used, but for this analysis, we restrict ourselves to the use of a simple random walk.

9.5.2.1 Random Walking

A simple random walk is a first-order Markov chain. A random walker starts from a given start node and selects one of the current node's neighbors as the next node by sampling based on the probability distribution of the edge weights. Let W_i be a random walk starting at vertex i walk in a fixed length of l, with the vertices in the walk represented as $(w_{i0}, w_{i1}, ...w_{il})$. While W_i denotes one walk starting from vertex i, we generate m such walks starting at vertex i.

9.5.2.2 SkipGram

In a given walk $(w_{i0}, w_{i1}, ...w_{im})$, we identify the vertices that appear in a window of length k as being in the neighborhood of each other. In the SkipGram architecture, the probability of the co-occurrence of the words in the window is maximized [172]. These co-occurring words within the window are positive pairs. A neural network with n inputs, a single hidden layer, and n outputs is used for learning embeddings. The hidden layer has $|e|$ neurons, which is also the dimensionality of the embeddings generated for each vertex. The output layer uses softmax. By the definition of softmax, $P(k|e_i) = \frac{exp(e_k.e_i)}{\sum_{j \in V} exp(e_j.e_i}$), for every vertex k in the neighborhood of vertex i.

9.5.2.3 Negative Sampling

The denominator $\sum_{j \in V} exp(e_j.e_i)$ in the softmax equation is computationally intensive. Deepwalk [202] uses hierarchical softmax to approximate this component, however node2vec [103] and word2vec [173] both use negative sampling. Through negative sampling, instead of reducing the weights of all the vertices not in the neighborhood of v_i, only a few of them are sampled and reduced.

9.6 Data Description

In this section, we describe the datasets used in our experiments. Table 9.2 shows the statistics of the number of minority, total, and the ratio of minority to total samples for each dataset used in the experiments.

9.6.1 Satimage

Satimage, taken from the UCI repository [74], is a popular dataset consisting of multi-spectral values of pixels in a satellite image [74] and is used in imbalanced classification problems [115, 171]. The task is to classify the central pixel in each neighborhood [74]. It is also part of the KEEL [10] dataset, a commonly used repository for imbalanced classification problems. Feature, sample, and imbalance information about this dataset is provided in Table 9.2.

9.6.2 Events

The Events dataset contains 70,533 observations about news patterns and the occurrence of terrorist attacks in each of the 51 U.S. states (including Washington, D.C.) over a period of 1,383 days. For each date and location, we synthesized 862 features from the Global Database of Events, Language, and Tone (GDELT), which monitors worldwide print, broadcast, and online news in over 100 languages [143].

TABLE 9.2

DATA DESCRIPTION

Dataset	# Attributes	# Minority samples	# Total samples	Imbalance Ratio
satimages	36	626	6435	0.0973
CA	862	24	1383	0.0174
NY	862	24	1383	0.0174
events	862	205	70533	0.0029

We additionally label each observation an event (1) if a terrorist event is recorded in the Global Terrorism Database [138] on that date in that state. If no terrorist event occurred, we labeled the observation a non-event (0). For the 1,383 days between March 18, 2015, and December 31, 2018, GDELT captured 858,969,588 records, of which 858,887,888 have traceable online sources. Thus, the Events dataset is primarily web-based and connects news events and themes to the occurrence of terrorist attacks in physical locations. We additionally utilize subsets of the Events dataset that are filtered based on a specific location, denoted by its two-character abbreviation. For example, "NY" consists of the observations across all 1,383 days, but only for the state of New York. For each observation, the date and location are treated as indexes only and are not included in the feature set.

9.7 Experiments

In this section, we report the results of experiments performed on the data sets described above. In all the experiments, we used logistic regression in conjunction with stratified three-fold cross-validation. We report performance using Area Under

the Receiver Operator Characteristic Curve (AUROC), which is a popularly used metric to measure the performance in imbalanced classification problems.

We generated embeddings of 32 dimensions for satimage and 128 dimensions for the other datasets CA, NY, and events, unless otherwise stated. The walk length l was set to 20, the number of walks per vertex m was set to 100, and the context window was k was set to a size of 10.

TABLE 9.3

BASELINES WITH LOGISTIC REGRESSION CLASSIFIER

	Satimage	Events	CA	NY
No Resampling	0.7245 (0.112)	0.6654 (0.760)	0.5354 (0.097)	0.5019 (0.068)
SMOTE [51]	0.7323 (0.105)	0.6112 (0.068)	0.5274 (0.116)	0.5116 (0.074)
ROS	0.7243 (0.112)	0.6000 (0.071)	0.5301 (0.0925)	0.5141 (0.077)
Poly-fit-SMOTE [95]	0.7347 (0.104)	0.6016 (0.072)	0.5389 (0.106)	0.5375 (0.077)
ProWSyn [25]	0.7380 (0.106)	0.6037 (0.068)	0.5327 (0.094)	0.5150 (0.066)
SMOTE-ENN [29]	0.7311 (0.106)	0.6123 (0.070)	0.6332 (0.025)	0.4994 (0.193)
SMOTE-Tomek [27]	0.7323 (0.105)	0.6112 (0.068)	0.5530 (0.100)	0.5117 (0.073)
RUS	0.7158 (0.099)	0.5944 (0.081)	0.4824 (0.098)	0.4269 (0.012)
Proposed method (knn)	0.9228 (0.004)	0.7311 (0.018)	0.6818 (0.038)	0.7109 (0.070)

9.7.1 Baselines

The classifier used to predict with the different representations in Logistic Regression. Various resampling methods have been included in this baseline, including oversampling (SMOTE [51], ROS, Poly-Fit-SMOTE [95], ProWSyn [25]), undersampling (RUS), and combined (SMOTE-ENN [29], SMOTE-Tomek [27]). We used the implementations of the python package imbalanced-learn [146] for SMOTE, ROS, SMOTE-ENN, SMOTE-Tomek, and RUS. For poly-fit-smote and ProWSyn, we used the smote-variants package [133]. For all the classifiers and resamplers, we selected the best parameters using grid search and reported the best performances of each method. As baseline, Table 9.3 shows the different AUROCs of different datasets. We see from the table that no method consistently outperforms the other methods in all the datasets, which is consistent with the literature [45, 217, 80, 164]. The baselines of CA and NY are, in particular, very close to to the performance of a random classifier. This task of predicting events is much harder when fewer samples are available for training the model.

9.7.2 ϵ-Neighborhood Graph Construction

Table 9.4 shows the results when using the epsilon neighborhood method with satimage. In this case, the pairwise distances between the graphs are computed with Euclidean distance. The threshold column shows the percentage of edges that are retained, shown in the # edges column. This percentage is used to calculate epsilon. When the percentage threshold is low, the graph is not fully connected. To show the increasing connectedness of the graph, we also report the number of connected components. The higher the number of connected components is, the more disconnected the graph is. We see that at a threshold of 20%, the graph is completely connected. As the epsilon threshold decreases, the number of edges increases. While the AUROC initially increases with increasing the number of edges, after a particular point,

TABLE 9.4

SATIMAGE ϵ-NEIGHBORHOOD

% threshold	# edges	epsilon	# connected components	AUROC
1	207947	3.45	222	0.8918 (0.133)
5	1035180	2.25	22	0.9212 (0.088)
10	2070152	1.68	3	0.9129 (0.0328)
20	4140279	1.19	1	0.9005 (0.0305)
30	6210418	0.99	1	0.8684 (0.0344)
40	8280558	0.86	1	0.8452 (0.0332)
50	10350697	0.75	1	0.7615 (0.0176)

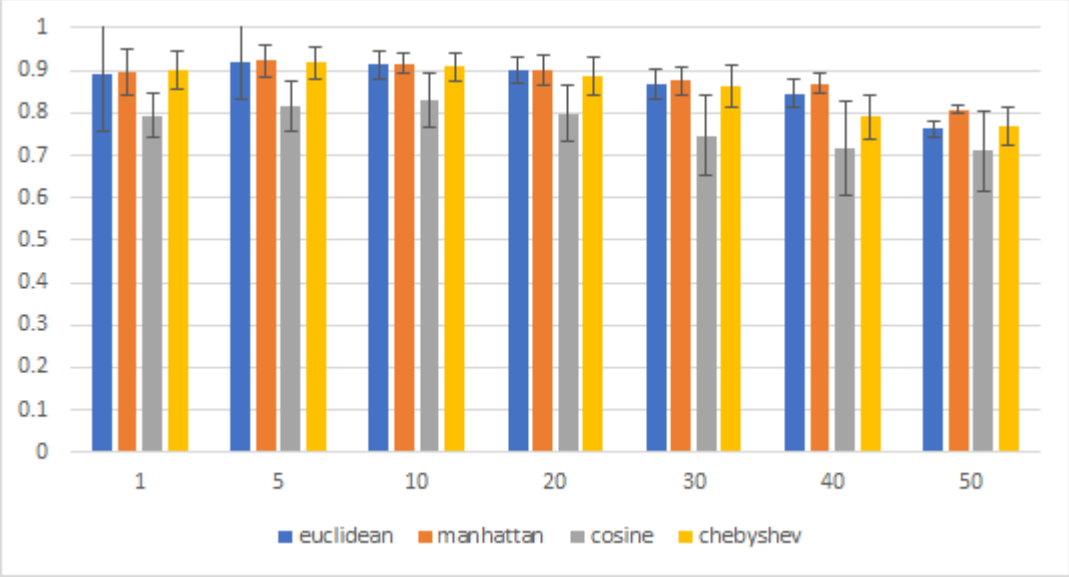


Figure 9.2. Different Distance Metrics on Satimage (AUROC on y-axis, % threshold on x-axis)

TABLE 9.5

CA ϵ-NEIGHBORHOOD

% threshold	# edges	epsilon	# connected components	AUROC
1	9556	0.342	6	0.2223 (0.0926)
5	47782	0.176	1	0.2751 (0.0228)
10	95565	0.166	1	0.3406 (0.0395)
20	191130	0.156	1	0.3536 (0.0895)
30	286695	0.15	1	0.5625 (0.0468)
40	382261	0.146	1	0.4851 (0.0212)
50	477826	0.142	1	0.5198 (0.0818)

it starts to decrease. Figure 9.2 shows the performance using the embedding features with the graph constructed using distance metrics of Euclidean, Manhattan, Cosine, and Chebyshev, with their definitions given in Table 9.1. While one single distance metric does not stand out as the best, cosine distance has the lowest performance in the satimage dataset. The number of disconnected components in the graph when constructed using Euclidean distance is similar to the other distance metrics. Due to this shortcoming of many disconnected components in the ϵ-neighborhood graph, we prefer to use knn for the graph construction phase.

From Tables 9.5 and 9.6, we see that even though the trend of AUROC is similar to what we see in the satimage results, logistic regression underfits on these embeddings. Note that the CA dataset is smaller than satimage and also more skewed in imbalance.

TABLE 9.6

NY ϵ-NEIGHBORHOOD

% threshold	# edges	epsilon	# connected components	AUROC
1	9556	0.338	3	0.4745 (0.0961)
5	47782	0.171	1	0.3667 (0.0762)
10	95565	0.163	1	0.2670 (0.0116)
20	191130	0.156	1	0.2779 (0.1004)
30	286695	0.151	1	0.3822 (0.0841)
40	382261	0.146	1	0.4123 (0.1352)
50	477826	0.142	1	0.6512 (0.0815)

9.7.3 KNN Graph Construction

We use euclidean distance for comparing the k nearest neighbors to the current vertex. Figure 9.3 shows the AUROC values for different values of k, whereas Figure 9.4 shows the results for CA and NY. We see that the knn construction of graphs leads to a generally improved performance in both NY and CA. From both the figures, we also see that the performance using knn is much more stable than ϵ-neighborhood. This is helpful because graph construction using knn is less sensitive to the parameter of k compared to the parameter of ϵ in the ϵ-neighborhood method.

In the knn method of graph construction, even though only k most similar vertices are used for drawing edges from the current vertex, the graph is not regular. Figure 9.5 shows the normalized distribution of vertex degrees in the graphs constructed from the satimage dataset. We see that the graphs constructed by knn are scale-free, in which few vertices are highly connected, whereas most vertices have a

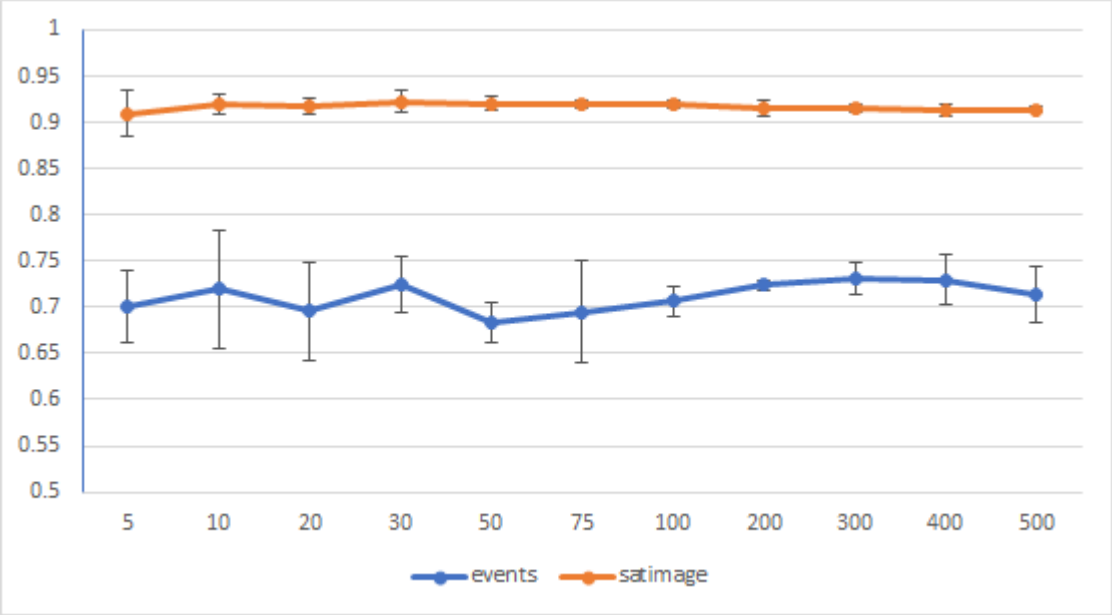


Figure 9.3. KNN Results for Satimage and Events (AUROC on y-axis, k on x-axis)

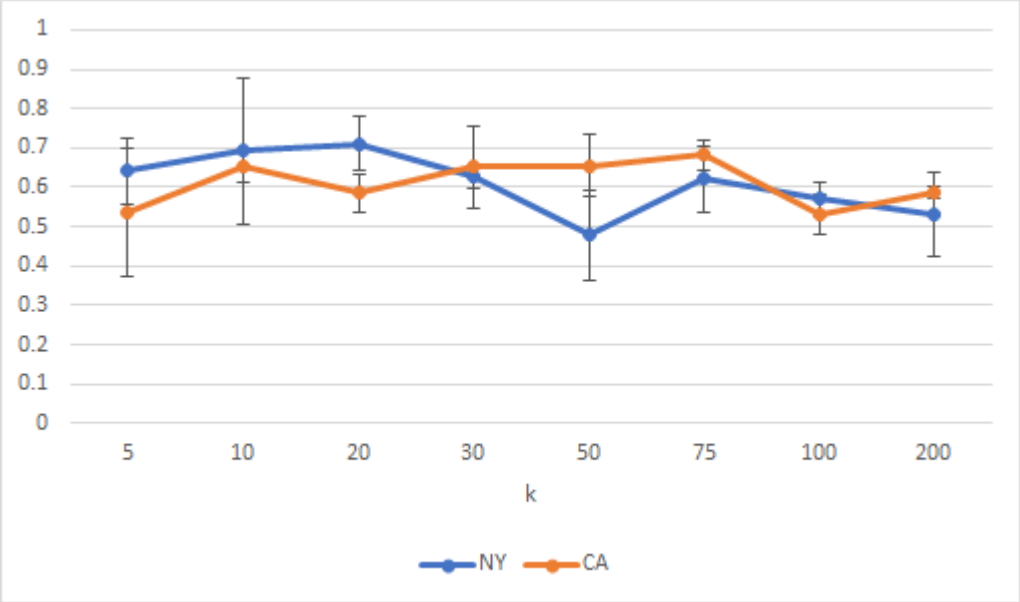


Figure 9.4. KNN Results for NY and CA (AUROC on y-axis, k on x-axis)

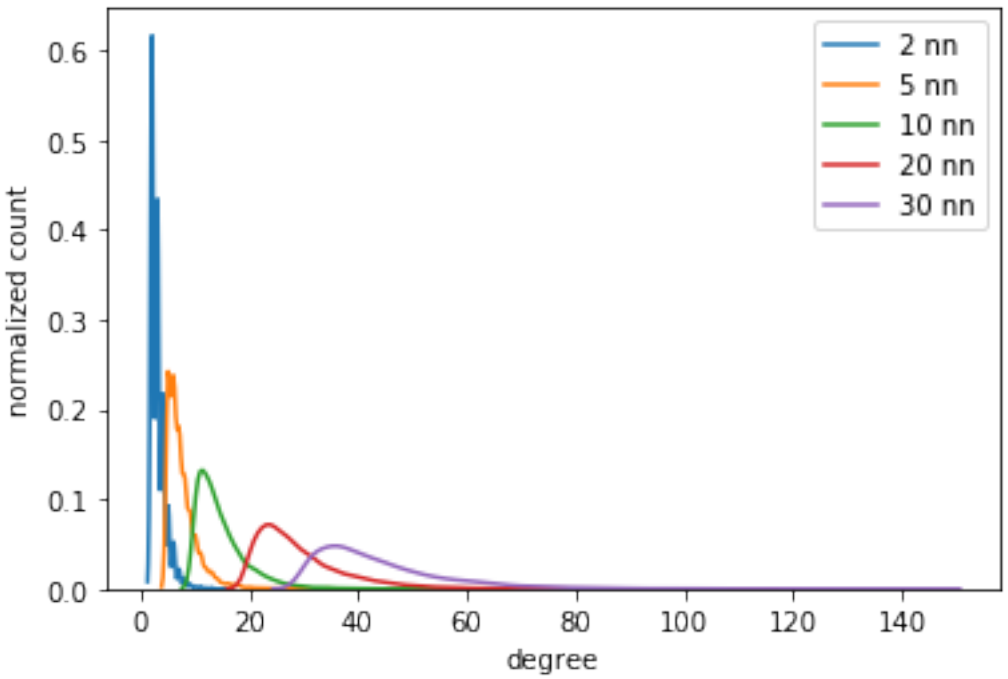


Figure 9.5. Degree Distribution for KNN Graphs (k plotted on the x axis, y axis shows the AUROC)

TABLE 9.7

KNN DEGREE DISTRIBUTION

	satimage			**events**		
k	mean	median	max	mean	median	max
30	44.0	41.0	145	31.9	31	86
50	73.2	68.0	206	58.1	52	1878
75	109.5	103.0	269	103.8	79	6643
100	145.2	137.0	334	150.4	106	10294
200	282.5	270.0	575	336.6	216	20090
300	412.2	396.0	741	521.4	327	26487
400	536.1	520.0	931	704.9	438	31237
500	653.6	628.0	1146	887.2	549	34871

TABLE 9.8

KNN DEGREE DISTRIBUTION FOR CA AND NY ONLY EVENTS

	events CA			events NY		
k	mean	median	max	mean	median	max
5	5.8	6.0	10	5.7	6.0	10
10	11.3	11.0	17	11.2	11.0	18
20	22.1	21.0	33	21.9	21.0	33
30	32.3	31.0	60	32.1	31.0	45
50	59.5	53.0	293	58.0	53.0	173
75	104.5	82.0	567	101.5	82.0	395
100	149.4	111.0	775	144.9	114.0	548
200	321.1	233.0	1275	310.5	236.0	1066
500	764.5	603.0	1382	736.2	633.0	1372

low degree. The degree distribution of the NY, CA, and events datasets with knn follow a similar trend. Table 9.7 shows some statistics to characterize the degree distribution of satimage and events datasets. We see that for the same value of k, which is the minimum degree in the graph, the mean, median, and maximum degrees are quite different in both the datasets. The statistics for NY and CA are in Table 9.8. We see that they are similar to those of the knn graphs constructed from the events dataset.

9.7.3.1 Sensitivity Analysis

Figure 9.6 shows a sensitivity analysis of various hyperparameters in this problem. The y-axis for all the plots is the AUROC. The top row shows the sensitivity on parameters for the events dataset, and the bottom row shows the same for the satimage dataset. The number of walks per node m denotes the number of walks starting from a specific node. In general, we see that this parameter does not affect the performance much. In the case of the events dataset, the performance for smaller values of the number of walks per node was slightly lower, but it quickly increased and stabilized.

In the cases of the length of walks l and window size k, we see a similar trend. In particular, events shows a slight drop in performance as these parameters increase. This drop is likely due to more and more irrelevant samples getting added to the context of the current sample as the window size or walk length increases. However, the changes in AUROC are slight, and we see that this parameter does not impact the performance much.

The embedding size in both the cases of satimage and events shows an increasing trend as the size of the embedding increases. Note that the x-axis is given on a log-scale for clearer visualization. The embedding size is the same as the number of neurons in the hidden layer of the skipgram architecture. Thus, an increase in the

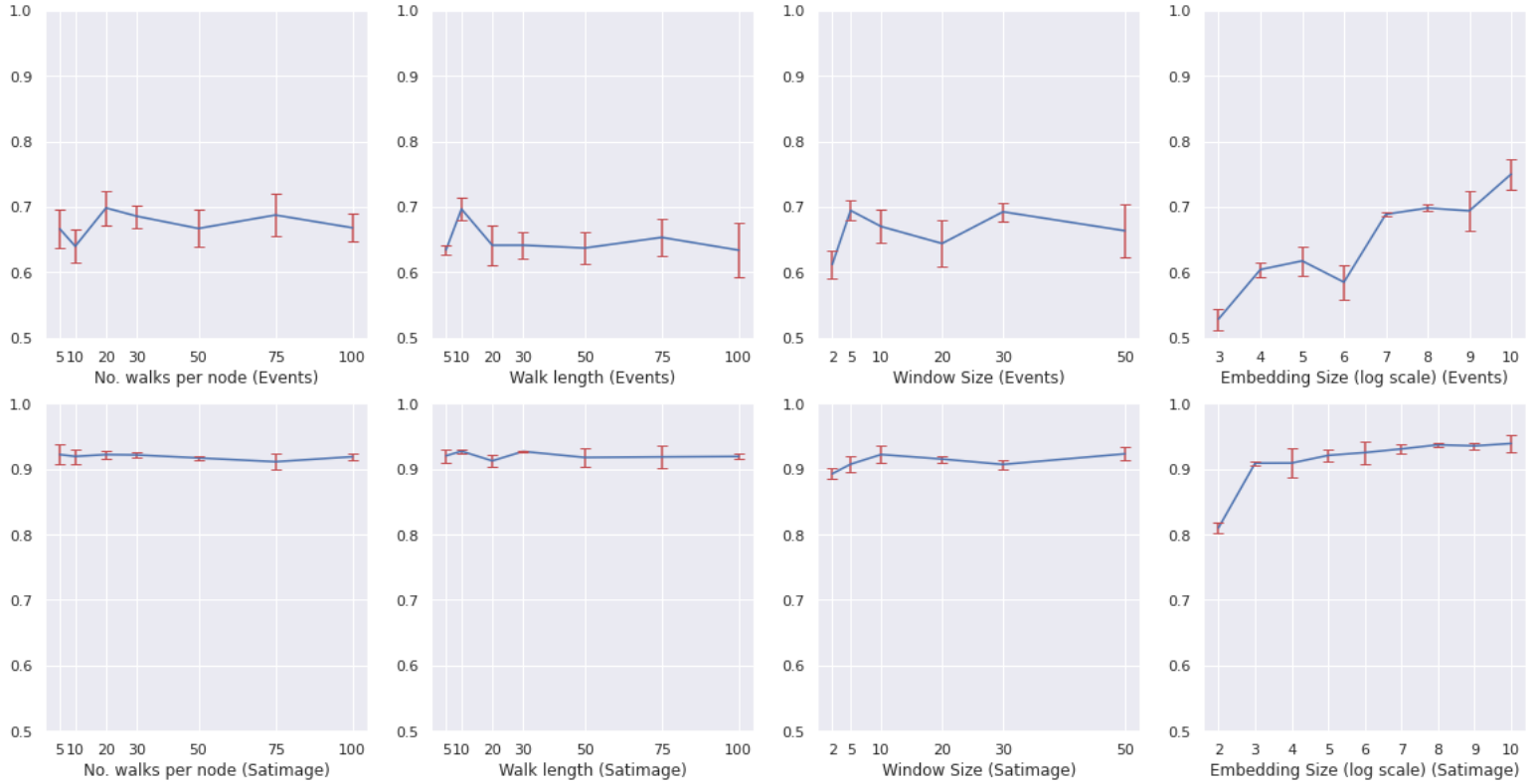


Figure 9.6: Sensitivity Analysis for KNN Graph Construction (y axis shows the AUROC)

number of hidden layer neurons would lead to encoding more contextual information. If we consider the embeddings of samples as a dimensionality reduced representation of the samples, a larger embedding size would be able to incorporate more information about the samples and their relationship to each other. An interesting point to note here is that even though the events dataset has 862 features, as can be seen from Table 9.2, increasing the embedding size from 512 to 1024 still shows a notable increase. In comparison, the effect of embedding size on performance shows the biggest jump for a smaller embedding size in the case of the satimage dataset and then increases slightly with embedding size. However, satimage has a smaller number of samples and feature size compared to the events dataset. We expect this behavior for larger embedding sizes with the events dataset as well. While these results would suggest increasing the embedding size as much as possible, the time complexity of skigram architecture is proportional to the embedding size as described in equation 5 of the analysis done by Mikolov et. al. [172]. Hence, the choice of embedding size is a tradeoff between performance and computation time.

Through this sensitivity analysis, we see that the effect of different hyperparameters on the performance is similar to those reported in random walk based embedding methods such as node2vec [103] and deepwalk [202]. In general, we see the robustness of these methods to hyperparameter tuning. Even for a small walk length and window size, we see better than random performance (AUROC of 0.5).

9.8 Discussion

Figures 9.7 and 9.8 show trends of different distance metrics used for constructing the knn graph. The definitions of the distance measures are given in Table 9.1. Both the figures show an increasing trend for all the distance functions as k increases in contrast to the trends of increasing ϵ-threshold, as seen in Figure 9.2. In the case of events dataset as shown in Figure 9.7, we see that Euclidean distance performs

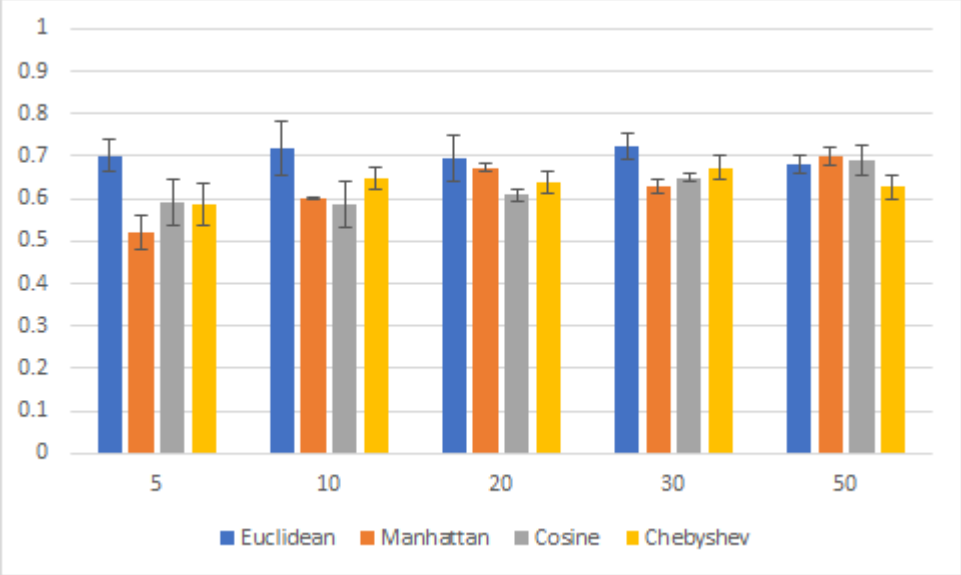


Figure 9.7. Using Different Distance Metrics to Construct KNN Graph in Events Dataset (AUROC on y-axis)

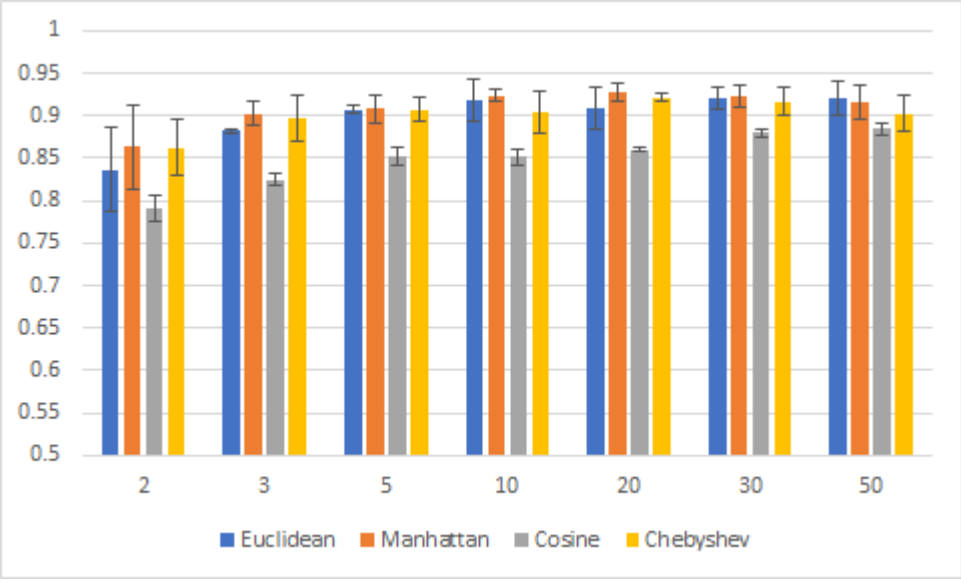


Figure 9.8. Using Different Distance Metrics to Construct KNN Graph in Satimage Dataset (AUROC on y-axis)

better than the other distance metrics for lower values of k, but this difference is not as prominent for higher k. Interestingly, the best distance measure is different for different values of k. For satimage, as shown in Figure 9.3, cosine distance still has the lowest performance in the knn graph construction method similar to the ϵ-neighborhood construction, as seen in Figure 9.2. Through these trends, we see that the choice of the distance measure used to construct the graph influences the quality of the embeddings for the imbalanced classification problem.

While knn graph construction improved the performance of our problem, it has certain shortcomings. As seen in Tables 9.7 and 9.8, even though knn graphs are constructed by connecting only the top-most similar nodes, the average degree of the graph is not usually k. This is because certain central vertices or hubs can be most similar to many other nodes and end up having high degrees. Many future methods have been proposed to deal with this problem, including b-matching, which aims to generate a regular graph. Whether a regular graph works better than the non-regular one remains to be seen in a future endeavor. However, these advanced methods have some practical issues such as the time and space complexity required to construct such a graph, and the available resources will determine the feasibility of these methods.

Another avenue worth exploring is the criteria used for graph construction. In this chapter, we use distance measures to judge the similarity of two instances. While distance-based methods are universal due to their applicability to any type of numerical feature vectors and robust across domains and sampling methods that generate the instances, it would be interesting to capture some of the generative processes in the distance function. By incorporating such a method, it could be possible to capture further nuances in the data while constructing a more useful graph. However, experiments need to be conducted to ensure the feasibility of this process in terms of computation time and space, since it requires the performance on the downstream

task as feedback to learn the best affinity matrix. In our case, that includes random walking over the graph, learning embeddings using a neural network, and then classifying the instances using these embeddings. Some methods in semi-supervised learning explore the combination of using both the original feature space and the new embeddings, as is done in Planetoid [257]. It is worth investigating the usefulness of such a combined approach.

Other modifications could be done to the embedding process to improve the representation of samples even further. For example, undersampling the majority class while generating positive pairs for the skipgram could help improve the runtime as well as not overrepresent the majority samples in the dataset. We investigated a balanced batch undersampling method based on the strategy described in [253]. This method considers a subgraph with majority class nodes randomly excluded in each epoch while keeping all the minority and unlabeled nodes. However, in the case of extreme imbalance, incorporating all the majority samples and learning embeddings for them while keeping the two classes balanced may require an unreasonably high number of epochs for training. An analysis of imbalance versus the number of epochs could help inform the tradeoff. Another issue is that subsampling the graph randomly leads to many disconnected components in the subsampled graph during each epoch, which generates possibly noisy embeddings for the nodes in the disconnected components.

9.9 Contributions

Many prediction problems have to perform classification on imbalanced data. In this chapter, we proposed the use of graph embeddings for improving imbalanced classification with graph embeddings. Across all of our datasets, the performance of classification using embeddings learned from knn graphs outperforms that using the raw features. This improvement suggests that embeddings learned on all the

data, including the training and testing samples, provided the classifier with more information to discriminate the two classes better. In our method, we used a simple random-walk based model for generating embeddings. We additionally compared two methods of graph construction: ϵ-neighborhood and k-nearest neighbors. From our experiments, we found that knn graphs lead to better performance than those constructed with ϵ-neighborhood, which was also noted in other literature [53]. We also observed that the performance using knn graphs was not as sensitive to the value of k as the ϵ-neighborhood graphs are to the value of ϵ.

While more sophisticated methods of graph construction can potentially be used to improve the performance even further, our results are promising for the use of embeddings in the task of imbalanced classification. The design of customized embedding generation algorithms that are intended for imbalanced classification specifically is a future avenue for exploration. Given the recent popularity of the field, many new embedding methods have been proposed with a sharper focus on the downstream application. Another future avenue is exploring inductive learning techniques for generating embeddings. In the transductive version, calculating the embeddings of new samples would require reconstructing the whole graph and then learning embeddings using the newly constructed graph. Future exploration leveraging inductive embedding techniques would be able to generate new representations for new samples without retraining the graph embeddings on all the samples. This would make the framework feasible for streaming applications such as fraud detection, where we pre-train the model based on existing data and predict the classes of new instances as they become known to the model.

In this chapter, we focused on random walking based embedding techniques due to their scalability in computation. However, recently graph neural networks have become popular for generating embeddings given an end-to-end application. For our classification task, we can use a graph neural network to generate new embeddings

of these samples. These embeddings would ideally be more specific to the task compared to the general-purpose embeddings generated by random-walking based methods. Thus, with all these possible directions, the use of embeddings in imbalanced classification problems has a promising potential for future endeavors.

Since imbalanced classification is a common problem, these techniques can also be used to better represent users when we only have access to their features and not a user-item bipartite graph that was used in the Chapter 8.

CHAPTER 10

UNIFIED REPRESENTATION OF NEWS AND SOCIAL MEDIA CONTENT

10.1 Overview

In this chapter, we return to the online consumption domain with a renewed focus on the content aspect of the user experience. In online content consumption, users are ultimately interested in the content provided by the services. Since users maintain similar topic interests across online platforms, content is valuable for user representation and can be used to link online news services with external data sources and websites.

Online news services employ strategies for personalizing and recommending articles to their users based on their interests. Before the era of the Internet, news outlets were a dominant mode of news consumption for most people. However, today social media is also a popular source of news information. Thus, news outlets have been using social media for news reporting as well as garnering more readers. With the advent of social media, there is now an opportunity to incorporate people's interests in and what they are saying about those topics in personalization and recommendation models. While this idea seems intuitively simple, there are many obstacles to be faced due to the two sources' disparate nature. In this chapter, we propose a framework to build a generalized graph of news articles and tweets that can be used for different downstream tasks such as identifying sentiment, trending topics, and misinformation, as well as sharing relevant articles on social media in a timely fashion. We evaluate our framework on a downstream task of identifying related pairs

of news articles and tweets with promising results. The content unification problem addressed by our framework is not unique to the domain of news, and thus can be applied to other problems linking different content platforms. This chapter is in the process of submission to a conference as a paper.

10.2 Introduction

One of the goals of online news providers is to improve customer satisfaction by recommending relevant articles in a timely fashion. To personalize content recommendations, news providers may collect user attributes such as demographic information or assess user interests through their chosen articles. However, when a user accesses a certain service for the first time, it is difficult to ascertain their interests. This is the cold start problem in recommender systems, and recent works have been leveraging social media to address it [142, 117]. Using social media for news personalization and recommendation has already shown promise in several works [231, 154, 17]

Before the era of the Internet, news outlets were a dominant mode of news consumption for most people, but recently social media has become a popular source of news information. Not only do people share news articles on social media, thus giving us insight into their topic interests, but they also discuss these topics through posts, comments, and reactions, providing insight into their sentiment and opinions. Thus, by combining news and social media data, there is now an opportunity to incorporate what people are interested in and what they are saying about various topics in personalization and recommendation models. This task of unifying news and social media, however, is not trivial due to the differences in language used in the two including formality, slang, memes, emoticons, length of text, and different intentions in communication. News outlets generally aim to inform and are not as biased as social media posts, which may be posted to convince others to adopt a particular opinion or express an idea or sentiment on a topic. Despite this bias,

in cases like hazard detection, political events, or crowd-sourced applications, social media platforms such as Twitter provide a gold mine of information that news may not be able to capture. This interplay between news and social media also allows us to study how social media affects journalism.

Besides differences in language, various technical challenges exist in accomplishing this task. In particular, Twitter imposes a limit on the number of characters that users are allowed to post. Thus, the context available to glean the tweet's topic is quite limited compared to news articles. Besides that, the use of different words to refer to the same concept or entity poses a challenge in inferring the topic of the tweet. On the other hand, while individual tweets are usually focused on one topic, each news article may cover multiple topics. Therefore, it is harder to detect the central topic to the article, which should be used to find the relevant tweet.

In this chapter, we propose a framework to build a unified representation of both types of content by using a graph of news articles and tweets to overcome these challenges. This generalized graph representation can be used for different downstream tasks such as identifying sentiment, trending topics, and misinformation, as well as finding relevant news articles to share on social media in a timely manner. We focus on using an entity-based framework to connect tweets to news articles. Recently, the use of entities has become popular for linking disparate sources of information; for example, Spitz and Gertz [223] use an entity-centric framework to detect new events and track them across multiple news sources. We then build a tripartite graph of news articles, entities, and tweets using various NLP techniques to represent the unified content space. We evaluate our unified representation on a downstream task of identifying tweets that are relevant to news articles. Through various evaluation measures, we see that our framework can retrieve related tweets better than the random baseline.

10.3 Related Work

The increasing use and prevalence of social media has changed how people communicate with each other and intertwined it with other media and types of content consumption such as news [43, 237]. This prevalence has also lead to citizen journalism [44] and ambient journalism [111] through social media. Furthermore, news organizations have also started using these social media platforms to promote their content and engage their users [112]. On the flip side, some studies have also reported on the use of social media for journalistic purposes such as news reporting [41, 197, 207]. In some cases, breaking news was first reported on social media like Twitter before mainstream news media had covered it [119, 237].

The advent of the Internet has also encouraged user-interaction and user-generated content in association with online news. Studies have found that incorporating user comments through forums improve the recommender systems [152, 19]. Trevisiol et al. [231] found that including the browsing behavior of users through referrer URLs improved the recommendation of articles by building a *BrowseGraph* and *ReferrerGraph*. Other studies have found a more direct connection between user content and news, for example, Tatar et al. [230] used the comments posted by users to rank news articles and infer their popularity, and Kourogi et al. [132] proposed a model that suggests attractive news headlines to share on social media.

The idea of using social media for online news personalization and recommendation is an established one and motivated by past research. Lin et al. [154] treat the opinions of social media influencers as auxiliary information in their news recommendation model and demonstrate the effectiveness of this method on the cold-start problem. Recently, recommendation frameworks also include social media preferences of users [17] and make news recommendations using users' social media information when available. Given all this intermingling of news and social media, it is worthwhile to explore a unified view of the two content spaces. Another benefit of creating

such a representation is that tweets are generally short and lack context. Thus, news articles can provide the required context to support NLP tasks on tweets such as topic modeling and opinion mining [106].

Some studies have proposed a unified framework to represent multiple news channels. For example, Mele et al. [170] use a topic modeling and Hidden Markov Model based approach for event detection and tracking through different news streams. Spitz and Gertz [223] use named entities to aggregate news from multiple streams. They also use a graph to represent all the content to support further downstream analysis. Similar to them, we use an entity-based framework due to the different styles of languages used in news articles and tweets.

The problem of linking news to tweets has been tackled in other studies. Guo et al. [106] use hashtags, named entities, or temporal constraints with a latent variable model Weighted Textual Matrix Factorization to link news with tweets. They use the title and a summary to represent the news article. Wang et al. [240] also propose a unified framework to find the most relevant news articles to a particular tweet by mining multi-aspect reflections. Another interesting and related problem tackled by Wei and Gao [245] is using tweets to summarize news articles. They find relevant tweets that share links to the news articles and use the text of the tweets as reference summaries for training their supervised learning model for news text summarization. The problem of generating relevant summarized social media discussion has also been tackled by Chakraborty et al. [49] wherein they use a network-based unsupervised approach to handle the noise and diversity of tweets. Li et al. [151] describe EKNOT, their framework that summarizes events using both news and social media perspectives. Their system presents a higher-level summary and overview of the events, while our framework attempts to unify the content representation at a granular level.

While Twitter has been linked with news for NLP tasks, it is also useful for answering questions about journalism and the relationship of news with social media.

Wihbey et al. [247] use Twitter to understand the relationship between journalists and social media. Tsagkias et al. [232] consider the task of finding republished articles on social media in the online reputation management domain, where organizations monitor their online reputation by leveraging social media. Republished articles could also generate new discussions around the topic. One of the applications of building a unified graph representation is to monitor the discussions surrounding news articles. Holton et al. [113] study the motivation of Twitter users behind linking news articles on Twitter. Hong [114] shows that the adoption of social media improves the online readership of newspapers. Kumar et al [137] predict which news articles will generate discussion on social media based on their content. Morgan et al. [181] explore the relationship between the perceived ideology of news outlets and the sharing of news on social media. Lehmann et al. [145] detect related discussion of tweeters after they tweet a particular news article. Bruns and Burgess [43] discuss some approaches that can be used to link news to twitter discussions with the help of keywords and hashtags, identifying temporal patterns and key users, and using graphs for analysis. We aim to support analyses such as these and future work in this area through the unified content representation of the two spaces generated by our framework.

10.4 Background

10.4.1 Named Entity Recognition

Since our framework utilizes entity-based techniques, we provide an overview of existing techniques in the next few paragraphs. Named Entity Recognition (NER) is defined as the task of extracting names of entities such as names of people, organizations, and locations from text [254]. This task generally consists of two steps: (1) the demarcation of the string in the text that is identified as an entity and (2) annotating the entity with its type, such as organization, person, location, and

time [26]. Recently, NER methods have started using deep-learning algorithms instead of feature-engineering based techniques. Yadav and Bethard [254] show that neural networks that infer features perform better than feature-engineering systems. NER methods such as a NER tagger provided by Stanford NLP toolkit [162], models that use LSTMs [139, 57] and conditional random fields [167, 218] are just a few of many NER models that have been proposed over the years.

Many works focus on identifying named entities in social media. In microblogs such as Twitter, the challenge of identifying entities is exacerbated due to noise, informal language, grammatical errors, lack of capitalization, and spelling errors as well as lack of sufficient context due to short message lengths [148, 153]. Limsopatham and Collier [153] use a bidirectional-LSTM and word embeddings to learn entities from tweets. Li et al. [148] propose a framework for identifying named entities in twitter using both the local context of the tweet and the global context through Wikipedia. In our framework, we also use both these contexts in identifying and linking the named entities. Efforts have also been made in entity annotation of the twitter corpus [66] by humans, a relatively more expensive undertaking compared to unsupervised models. The task of entity recognition has useful applications to governments and companies such as hazard detection and early crisis response [148]. Further, the quality of entities detected can be improved by linking them to a knowledge base. For example, Yamada et al. [255] link the entities to Wikipedia to improve the identification of entities in twitter.

10.4.2 Named Entity Linking

Since we wish to connect tweets to news articles by linking entities, each source's text needs to be linked to the correct entities. This brings us to the problem of disambiguation and aliasing. Entity disambiguation refers to the task of linking entities when multiple of them share the same name but refer to different entities.

This commonly occurs when multiple people share the same name. For example, there exists a poet and an Olympic gold medalist; both are different people with the name Kevin Young. However, when an article refers to the poet, it should be connected to a separate entity than when it links to the athlete. The Wikipedia disambiguation page for Kevin Young shows seven different people at the time of this writing [4]. Another related problem is aliasing, in which a particular entity could be referenced in multiple ways. A common example of this is a person's name, which could be written in different formats, including the first name and last name, initials only, the last name only, etc. An example of the wikidata articles on Donald Trump shows 14 aliases in English alone [2]. In general, the steps for entity linking are as follows: (1). Use a Named Entity Recognition system to identify entities in a text, (2). Generate a set of candidate entities using a knowledge base such as Wikipedia, (3). Rank the candidate entities using methods like prior probability and the context of the text in which the entity is present, (4). Select the most likely entity from the candidate set as the linked entity.

Given that tweets not only use different ways of referring to an entity, but also include additional complications due to non-standard language and spelling errors, many works have explored the problem of entity linking in twitter. Basile and Caputo [26] provide an overview of entity linking methods to be used specifically for tweets. Urata and Maeda [234] use Wikipedia for word-sense disambiguation of entities in tweets. Waitelonis and Sack [238] use a DBPedia knowledge base to link entities in tweets. Thus, we also link entities in our framework to knowledge bases, so that we can take advantage of the context and prior probability for entity disambiguation and aliasing.

10.4.3 Coreference Resolution

While these methods aim to learn entities from tweets instead of the general news domain, our problem requires us to learn entities from both the news and twitter domains. Most tweets have only one or two entities since each tweet is focused on one topic usually. However, identifying entities from news provides the opposite challenge. News articles tend to have many entities, including locations, dates, and times along with persons and organizations. However, for the task of connecting these news articles with tweets, many of these entities are irrelevant to the tweet, and we need to find the important ones.

One way of dealing with this problem is by using coreference resolution. Coreference resolution is the task of finding all the references in a text made to a particular entity occurring elsewhere in that text. For example, pronouns typically refer to some entity in the sentence. The task of coreference resolution is to identify which pronouns are related to which entities in the sentence and cluster them correctly. Modern methods rely on deep neural networks as they perform better than syntactic parsers and feature-engineering based methods [104]. The various coreference resolution models can be broadly categorized into mention pair classifiers, entity-level models, latent-tree models, mention-ranking models, and span-ranking models [104, 140]. The models proposed by Clark and Manning [61, 60] are examples of mention-ranking models. The models by Lee et al. [140] and Gu et al. [104] are examples of span-ranking models. In our framework, we use a mention-ranking model that is described in further detail in the next section.

10.5 Data Collection and Preprocessing

In this section, we describe the process of acquiring data used in the construction of the twitter-news graph.

10.5.1 Twitter Data Collection

The New Yorker (TNY) often tweets articles, which are associated with The New Yorker's Twitter handle "NewYorker". This gives us a good starting point for identifying tweeters who are engaged with the New Yorker content. By analyzing the content that these users generate, we would be able to gain a better understanding of how potential users and subscribers engage with the New Yorker content and what their interests are. This is similar to the strategy used by Nigam et al. [186]. Since we are specifically looking for tweets relevant to the New Yorker articles, we collected tweets containing keywords, including the "New Yorker" and streamed tweets of users who were more engaged with tweets generated by the New Yorker twitter handle. By streaming the data, we hope to capture a more complete and representative version of Twitter with respect to TNY. Thus, the steps we used for collecting tweets are as follows:

1. Collect the most recent New Yorker tweets from 12th December 2019 to 3rd January 2020. This resulted in 765 tweets by NewYorker
2. Sample a subset of 2,257 users that retweet NewYorker tweets. Since NewYorker has many followers and likes on each tweet, we selected tweeters through retweets with the expectation that these users are more engaged with NewYorker than followers and users who like the tweets.
3. Stream tweets from 12th December 2019 to 3rd January 2020, based on the following criteria. Stream 1,928,699 tweets generated by users selected in step 2 and 765 tweets by NewYorker. We also stream tweets containing keywords including "New Yorker" and various author names who frequently write for the New Yorker, resulting in a total of 3,006,233 collected.

10.5.2 TNY Data Collection

Since we aim to align TNY articles with tweets, we collected content data of articles that were clicked on during the same time period that the tweets were collected. This data was directly obtained from the clickstream log of users who accessed TNY articles within the time frame that the tweets were collected.

10.5.3 Text Preprocessing

We used the same data preprocessing steps to normalize tweets and TNY articles. Normalizing tweets is not a trivial task due to the informality of language used, including the use of slang, emoticons, and spelling errors, and many research efforts have been made to improve this process. However, this is an important step in our framework because news articles use more formal language. With our aim to unify the text of tweets and news articles, we need to apply certain preprocessing steps such that the informality of twitter text is reduced. User names, URLs, numerical values, including date and time, and email addresses were replaced with a placeholder such as ¡email¿ or ¡url¿. Contractions such as "can't" were expanded, hashtags were separated, and emoticons were replaced. We used *ekphrasis* [30] for this data processing step, as they provide a comprehensive library for cleaning the text data and are geared towards text from social media. The following are the steps we used for preprocessing and tokenizing the data:

1. Filter out tweets/articles that are not in English. Since the entities we use are from English corpii, this is an important filtering step for gleaning context.
2. Eliminate duplicate tweets/articles in the corpus. We represent this original corpus of unique texts as $T_{org-twitter}$ and $T_{org-tny}$ for tweets and news articles, respectively.
3. Normalize 'url', 'email', 'percent', 'money', 'phone', 'user', 'time', 'date', 'number'
4. Annotate hashtags, elongated words, emphasized words and censored words in the text.
5. Unpack hashtags such that each word in a hashtag is a separate token. Expand contractions in the text.
6. Using a dictionary, identify and replace emoticons with words in the dictionary. We denote these preprocessed lists of tokens as $T_{norm-twitter}$ and $T_{norm-tny}$ for tweets and news articles, respectively.

   We use these preprocessed lists of tokens for content representations. However, to identify entities in the text, we use raw data, as punctuation and capitalization are essential components for identifying named entities.

10.5.4 URL Preprocessing

To establish direct connections between news articles and tweets, we parse URLs in all the text. Since URLs can be shortened or have aliases, we first parse the URLs to generate the full-URL. These preprocessed URLs are used for creating connections between tweets and news articles. For this chapter, we only consider links mentioned on tweets that are direct URLs of the New Yorker articles. While we could potentially align matches between tweeted URLs and referrer URLs in the clickstream data, it may lead to spurious connections, especially in case of many-to-one mapping between referrer links and article links. However, with some additional preprocessing, we could incorporate this information, and expand on this idea in our future work.

10.6 Framework

In this section, we will explain the various components of the unified graph model. We will also explain the motivation behind various design choices.

10.6.1 Named Entity Representation

Each text document in the corpus is represented as a list of named entities that appear in the tweet or news article. We run the Named Entity Recognition (NER) parser through $T_{org-twitter}$ and $T_{org-tny}$. We use spacy's NER tool [116] to extract entities. Spacy's Named Entity Recognizer uses a convolutional neural network [195] and is trained on OntoNotes 5 corpus and recognizes 18 entity types. However, we only track entities of types PERSON, NORP, FAC, ORG, GPE, LOC, PRODUCT, EVENT, WORK_OF_ART, LAW, and LANGUAGE. We do not include numerical entities, including date and time, since they generally do not provide useful links between two documents. For example, if one text mentions "two" apples, and another tweet mentions "two" cycles, connecting these texts by the word "two" leads to noisy

edges in the graph.

Figure 10.1. Examples of Linked Entities

10.6.2 Linked Knowledge Base

While entities were able to provide us with useful edges between text documents, two entities with the same name do not necessarily refer to the same entity. For example, different people with the same name could be mentioned in different text documents, but they would get mapped to the same entity. On the other hand, the same entity could have multiple representations in the corpus. We tackle this

problem by using Named Entity Linking and Disambiguation. Entities are linked to a knowledge base, Wikipedia and Wikidata, in our case. Spacy [116] has provided a fast implementation [1] of entity linking with the following steps:

1. Input NER mention from the text and generate candidate entities for each mention from the knowledge base.
2. From text, embed the sentence context s_i and entity type of the mention t_i.
3. For each candidate entity from the knowledge base, calculate the prior probability p_i and encode the entity description d_i.
4. Concatenate s_i, t_i, p_i, and d_i into a single vector and learn the probability of the entity given the mention.

Figure 10.1 shows examples of the ten most frequent entities that were aliased in the tweets. Not only did we find different variations of the name Donald Trump, but we also discovered that Eustace Tilley is an alias for The New Yorker and the tweet containing it was linked to the entity corresponding to The New Yorker successfully. An example of

10.6.3 Coreference Resolution

For coreference resolution, we use the implementation provided by Hugging Face [2]. Their implementation is based on the mention-ranking model by Clark and Manning [60, 61]. Mention-ranking models use the likelihood of coreference to score pairs of mentions [60]. The steps for mention-ranking model are: (1). Extract mentions from the text, (2). Compute a set of features for each pair of mentions, (3). Using the features, find the most likely antecedent for each mention. Return clusters of mentions. In [61], they use a learning-to-search to train a neural network to merge clusters, whereas [60] uses a reinforcement learning algorithm to optimize the model

[1] https://github.com/svlandeg/spaCy/tree/3fbab231b530e6b638c3443cf37c38c62d0e4647/bin/wiki-entity-linking

[2] https://github.com/huggingface/neuralcoref

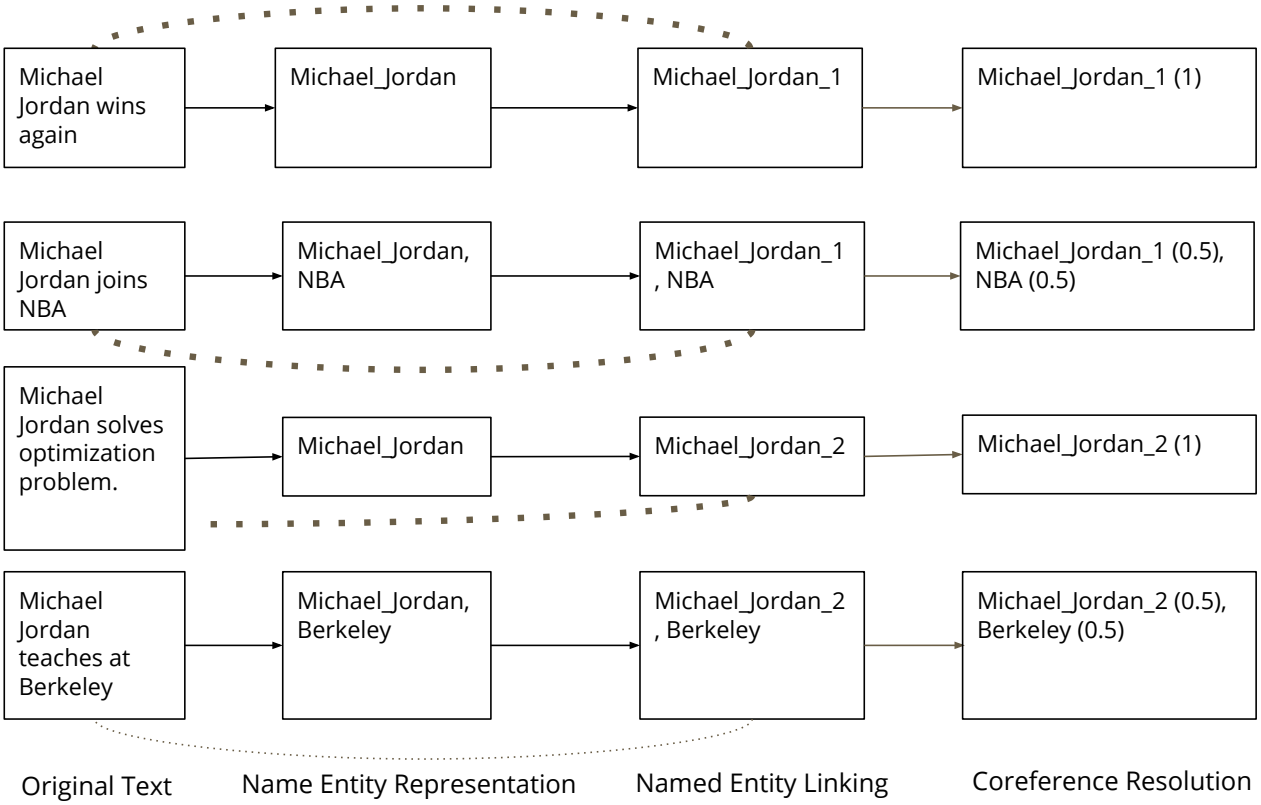


Figure 10.2: Illustration of Different Stages

for coreference evaluation metrics. Figure 10.2 shows an illustration of various stages on some sample text.

10.6.4 Graph Construction

We construct a tripartite graph with the first layer being news articles, the second layer being named entities, and the third layer consists of tweets. We restrict the entities to the top 1000 most frequent entities among the tweets. The set of edges are drawn between tweets and entities E_{T-E} based on whether the tweet contains the entity. The set of edges drawn between the entities and news articles are denoted as E_{N-E} and drawn in a similar fashion. The set of edges between news articles

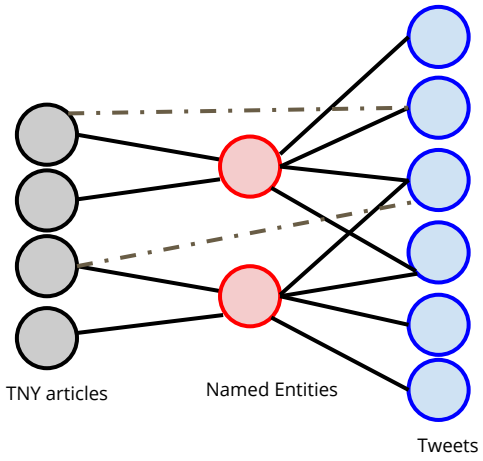


Figure 10.3. Tripartite Graph Schema

and tweets E_{N-T} are drawn if the tweet directly links the news article in it. These edges are the rarest in the graph. Figure 10.3 shows a diagram of such a graph. The set of edges E_{T-E} and E_{N-E} are weighted using coreference resolution. The edge is weighted by the size of each coreference cluster in the text. All the E_{N-T} edges have weight 1.

10.7 Graph Description

From all the tweets, we extracted 1,964,367 entities with 68,621 unique entities. Of all the entities, extracted, 55,376 occurred in both tweets and news datasets. Building a tripartite graph with all of these entities led to 460,485 connected components in the graph, with the largest connected component having 392,450 nodes and 710,427 edges. Thus, we have a sparsely connected graph. The number of direct references made from tweet to news article, i.e. E_{N-T} is 53, a low number compared to the other types of edges in the graph. In the tripartite graph, we only keep the most frequent 1000 entities and retain 21,518 news articles and 369,880 tweets.

Figure 10.4 shows a normalized histogram of the log count of the number of

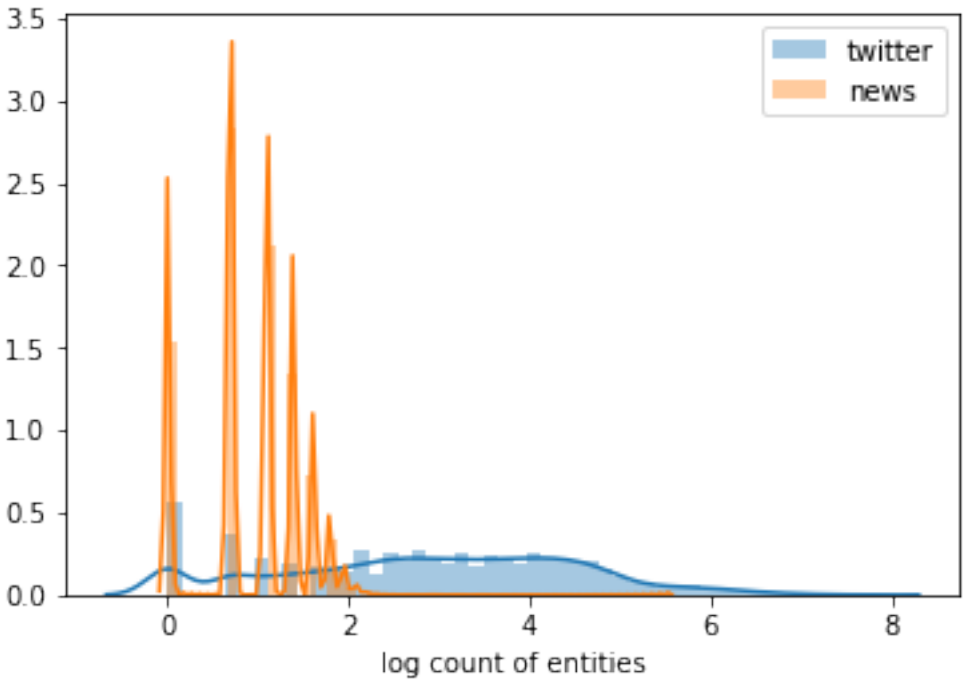


Figure 10.4. Normalized Histogram of the Log Count of Entities in Tweets vs News Articles

entities in each document (tweet or article). We see that news articles have more entities in them compared to tweets. In fact, tweets have 1.9 ± 2.0 average entities in a tweet, with the median being 2 entities per tweet. In contrast, news articles have 57.0 ± 117.6 entities on average, with the median being 19 entities per article. In the tripartite graph, of the three groups of nodes, the entities have the highest degree distribution, followed by the news nodes, and then tweets. The degree distribution of the nodes also follows a power-law distribution, with the maximum degree being 49,853, but the mean is 3.6 ± 120.0, and the median is 1. Thus, while the degree of entity nodes can be extremely high, many tweets are only connected to one entity.

10.8 Article-Tweet Relatedness

The annotation of semantic relatedness was one of the tasks in the SemEval-14 challenge [163], wherein participants submitted a system that could rate the relatedness of two sentences. We propose a task similar to this for evaluation, where we evaluate our framework on the task of finding tweets most similar to news articles and rating their relatedness. We generate embeddings for all the nodes in the constructed graph using a simple random walker and skipgram architecture with negative sampling, the same as node2vec [103] with p and q set to 1. Then for each news article, we find the k most similar tweets by calculating the cosine similarity between each pair of news articles and tweets embeddings. We rank the top 100 news articles with the highest similarity to tweets. For each news article, we report the top k most similar tweets.

10.8.1 Baseline - Random Matching

In this subsection, we describe the random baseline we use to compare the performance of the framework with other methods to understand how well it works. In this baseline, we randomly match tweets to news articles. This baseline is supposed to give us an idea of the volume and diversity of topics in tweets. Since many of the tweets are related to politics, the relatedness of a tweet and news article may not be perfectly random.

10.8.2 Evaluation

We consider various methods of evaluation, using both automatic and human evaluation methods as described below. Human evaluation is important to get the subjective perspective of text relatedness. However, it is expensive, so we also use automatic evaluation methods based on n-gram matching.

10.8.3 Amazon Mechanical Turk

We use Amazon Mechanical Turk to rate the relatedness of a tweet and a news article. We use this on the random baseline and the full framework. Workers were asked to rate whether a tweet and a news article were related or relevant to each other. The annotators could select out of three options: (1). the news article and tweet are completely unrelated, (2). the news article and tweet are broadly related, and more context is needed, and (3). the news article and tweet are definitely related. The news article was represented as the title, followed by a summary of the article with at most 50 words. The summary was generated using TextRank, an algorithm that ranks sentences using the PageRank algorithm. Each pair was assigned 3 unique mturk workers. We annotated the 30 most similar with 5 tweets each, thus resulting in 150 unique pairs. The mean and standard deviation of the ratings on the full framework is 1.87 ± 0.73 with a median of 2, which means that most workers have assigned 2's. The raw counts for the full framework are 181 for 1 - there are completely unrelated, 193 for 2 - there is a possibility of being related, and 105 for 3 - they are definitely related. In comparison, the counts for the random pairs are 262 for 1 - they are completely unrelated, 54 for 2 - there is a possibility of being related, and 134 for 3 - they are completely unrelated. The mean and standard deviation of annotations for the random baseline is 1.72 ± 0.89, with a median of 1 rating. Note that this task is challenging due to the news articles being summarized for the workers. The workers took a median time of 28 seconds to finish their tasks.

An example of a related summarized news articles and tweet pair is shown below:

News article:- Jana Prikryl Reads Anne Carson: Jana Prikryl joins Paul Muldoon to read and discuss Anne Carson's "Stanzas, Sexes, Seductions," and her own poem "Thirty Thousand Islands."

Tweet:- @VChangPoet I keep mentioning this every time someone asks about an amazing book, but I'll do it again: Anne Carson... https://t.co/XHWQXfCsaW

10.8.4 Automatic Evaluation

For automatic evaluation we consider different strategies based on n-grams. Scores such as BLEU and ROUGE are n-gram based evaluation measures that can be used for the task of text summarization, in which a long text is summarized to a shorter version with fewer sentences. These evaluation measures require a gold standard reference summaries provided by humans with which to compare the target summary. Humans provide reference summaries for use with BLEU and ROUGE. We can think of the related tweets as summaries of the news articles to which they are related. Thus, we use the sentences of the news article as references for the related tweets. In other words, if one of the tweets that a news article was paired with was a sentence from the article, the BLEU and ROUGE score would be the highest. While we do not expect high BLEU and ROUGE scores, we expect there to be at least a few n-grams in common which can be matched and use these measures to compare the different methods.

TABLE 10.1

AUTOMATIC EVALUATION OF TEXT RELATEDNESS

Model	BLEU	ROUGE-1	ROUGE-2	ROUGE-3	ROUGE-4	ROUGE-L	ROUGE-W
Full framework	0.0112	0.0682	0.0075	0.0013	0.0003	0.0907	0.0354
Framework - coref weighting	0.0056	0.0717	0.0084	0.0016	0.0003	0.0939	0.0356
Full framework using Stanford NER	0.0073	0.0573	0.0017	2.88E-05	0	0.0785	0.0300
Full framework - KB linking	0.0073	0.0682	0.0073	0.002	0.00098	0.09133	0.0355
Random baseline	0.0025	0.0405	0.001	3.08E-05	2.02E-07	0.058	0.0221

10.8.4.1 Bilingual Evaluation Understudy (BLEU)

: BLEU [191] is a precision-related measure [155]. While it was originally proposed for the machine translation task, it has also been used for text summarization [120]. We calculate the BLEU score in Table 10.1.

10.8.4.2 Recall-Oriented Understudy for Gisting Evaluation (ROUGE)

ROUGE is a metric used for evaluating automatic summaries using human-generated summaries as references [155]. ROUGE-N measures the n-gram recalls between the target summary and reference summaries generated by humans. ROUGE-L measures the longest common subsequence whereas ROUGE-W measures the weighted longest common subsequence between the target and reference texts. Table 10.1 shows various ROUGE scores.

We see from the tables that the full framework performs the best in BLEU and second-best in ROUGE. The framework performs better than the baselines, and we have also evaluated the performance through ablation studies by removing different components of the framework. We also include a comparison with a framework that uses StanfordNER [83] for entities and see that it does not outperform the spacy NER.

10.9 Contributions

In this chapter, we addressed the problem of unifying content spaces from different platforms, namely news and twitter, by proposing an entity-based graph representation. We used different NLP techniques in the graph construction process, including named entity recognition and linking and neural coreference resolution. We evaluated the graph on one downstream application, in which we retrieved the most related tweets to a particular news article. We showed that the framework returned more

relevant tweets than the baselines. While these results are promising, this approach holds even further potential for improvement.

We hope to address certain limitations of this proposed framework in the future. In our graph construction process, we did not link tweets using retweet information and other twitter metadata, but that could potentially add more signal in the graph, thus improving the framework. Due to the lack of context, we are unable to leverage more sophisticated contextual embedding algorithms such as ELMO [203]. However, by aggregating tweets through clustering or combining retweets, we may be able to leverage more contextual information in the graph construction process, such as weighting edges.

Another direction of exploration we would like to consider in the future is incorporating more downstream applications. We evaluated our method on one task of text relatedness. We saw that while we were able to retrieve many relevant tweets successfully, there is still an error component. To support downstream applications such as sentiment analysis and opinion mining, it would be better to reduce the error rate even further. We would also like to answer research questions related to journalism and user interests that this graph would be able to support.

The content unification problem addressed by our framework is not unique to the domain of news, and thus can be applied to other problems linking different content platforms. For example, a unified representation of academic papers and social media would help give insights into scientific outreach and how public interest influences scientific progress. Given the promise shown in this work, our framework has the potential to benefit many downstream applications that require the unification of content across platforms.

Thus, by linking news content and social media, we can improve the user experience further by providing recommendations based on trending topics on social media. If we have access to social media information of users, it can be used to represent

users and learn their characteristics. This is particularly useful for new users because we they have no behavior on the website, and neither have they consumed any content. In this case, externally generated data such as social media is valuable. Finally, we can gain deeper insights into the news content by following the narrative around it on social media, which can be used for personalizing the news recommendations to users and help improve their experience even further.

TABLE 10.2

SUMMARY OF TOOLS AND TECHNIQUES

Problem	Features defined	Techniques used
MOOC	Behavioral features using video clickstream - engagement, interactivity, reflectivity, and impatience Clustering individual emotions into higher-level categories called quadrants. Positivity score as a moving average of individually reported emotions.	Statistical analysis (ANOVA, co-occurrence analysis, transition likelihood, correlation, interrater agreement using Cohen's Kappa), K Means Clustering, PCA for visualization
FYS	Defined achievement ratio to normalize the calculation of grade change	Statistical analysis (Odds Ratio, Correlation, Mann Whitney U Test)
Gender prediction using content	User profiles using content, topic profiles to represent users	Representing text through bag of words and topic modeling, resampling using ROS, RUS, SMOTE, and SMOTE variants (hyperparameter tuning done using grid search), XGBoost classifier using topic modeling
Demographic prediction	Representing users with graph embeddings	Statistical analysis (hypothesis testing for behavioral features, correlation), various classifiers and regressors, node2vec
Using graph embeddings for imbalanced classification	Represent samples with graph embeddings	Different graph construction methods, node2vec, statistical analysis on degree distribution (histograms),
News-Twitter Content Unification	Tripartite graph of news articles, tweets, and entities	Topic modeling, word2vec, doc2vec, NLP techniques (named entity recognition, linking, coreference resolution), text relatedness using graph embeddings and cosine similarity, node2vec

TABLE 10.3

SUMMARY OF CHALLENGES

Problem	Data challenges	Solutions
MOOC	Emotions are noisy, conflicting, spontaneous and individual emotions are categorical (challenging representation for analysis and modeling), small ratio of students who did the surveys in order, identifying implicit sources of emotions irregular time series of emotion sequences	Define emotion quadrants to reduce the number of categories, filtered students on the correct survey order, sentiment analysis on discussion forum using word-affect lexicon. Defined positivity to reduce the effect of noise. Convert emotions to a numerical representation using a word affect lexicon. Define a fixed length feature vector to represent valence sequences.
FYS	Small ratio of students struggle (imbalance), identifying which features of data collected through different platforms can be used in real-time to identify struggling students, no control group	Used a non-parametric pre-post test since target variable is not uniformly distributed. Used historical data for comparison and to compensate for lack of control group.
Gender prediction using content	Numerical resampling techniques not better than baseline, resampling using GAN was leading to mode collapse	Used text-level resampling with SeqGAN which generated tokens instead of numerical data
Demographic prediction	Heterogeneous features with various methods of representation, some of these representation methods performed similar to each other, high dimensionality with URL features and bag of words, data sparsity	Generated extensive sets of features from content, behavior, and combined for comparison using different models, used graph based approaches for data sparsity, reduced dimensionality of URL features based on popularity, graph embeddings were robust to both types of data sparsity
Using graph embeddings for imbalanced classification	Different graph construction methods lead to different results, graph construction step does not scale well with number of samples	Identified parallelizable graph construction method (KNN), and tested for robustness to hyperparameters, analyzed descriptions of constructed graphs to identify improvements
News-Twitter Content Unification	Size of tweets is short compared to news articles so methods combining the representation of both eg. topic modeling, word2vec, and doc2vec suffer in performance. Out of vocabulary words an issue for word2vec Topic modeling produced coherent topics, but not a general representation. Tweets are noisy, contain emoticons, spelling errors, slang	Aggregate tweets by author to get longer text documents for topic modeling, use ELMo instead of word2vec for OOV words, use graph-based representation for a general representation. Use entity-based NLP techniques that are more robust to noise

CHAPTER 11

CONCLUSION

Through this dissertation, we have explored the topic of understanding user characteristics in the context of content and behavior. We studied this topic in two domains - learning analytics and online news consumption. We have also considered the challenges of user representation and imbalanced classification, both in the context of predicting users' characteristics and in the traditional supervised learning setting. Table 10.2 provide a summary of the various techniques used for these problems in this dissertation. Table 10.3 highlights some of the challenges faced during the data exploration stage, along with solutions to overcome them. To overcome various challenges, we also defined features as highlighted in Table 10.2.

In the domain of learning analytics, we have analyzed students both in the online and offline settings. In the online environment, i.e., the MOOC, we saw that students' emotions were related to their behavior and performance in the class. Table 10.3 outlines some of the data challenges encountered and solutions used to tackle them. Since students reported the emotions through different platforms, there was heterogeneity even in the sources of emotions. Surveys had categorical emotions represented through words, SAMs had emotions represented on a scale, and the emotions in the discussion forum were inferred through text. The solution that we used for this problem was converting all of them to the same numerical valence scale. Other challenges with emotion data are that they are spontaneous, conflicting, and noisy. For robustness to this issue, we defined positivity as the moving average of positive valences.

In the First Year of Studies course, we were not only able to use students' behavior (whether they submit their homework) and performance (their grades on the assignments) to identify students who are struggling in the class. Not only did we identify them, but we also intervened with them to improve their performance. An analysis shows that the intervention was effective at improving the performance of students who were struggling. In this work, we had a needle-in-a-haystack problem of identifying struggling students in this Mastery based course. One of the practical challenges we face d was identifying useful data for this problem from the variety of data collected and stored on different platforms. This task was particularly challenging as we needed to collect and identify students in real-time and was made even harder due to different types of missing values, including students and instructors not submitting assignments or grades on the correct platform in a timely fashion. Thus, we focused on easier to track features that could be gleaned immediately from the clickstream data, such as assignment submission.

In the online content consumption domain, we explored different ways to characterize the user, including content, behavior, and combinations of the two. The content-based representation uses the articles' metadata, particularly the text that the user consumes when they perform the clicking activity. The behavior-based representation uses only the information available in the clickstream data, such as the URL clicked on, location information, device and browser information, and timestamp. We investigated various methods to represent the user, including feature vectors, topic modeling, bag-of-words, as well as a graph-based method that represents users through embeddings. While many of these methods were useful, the graph-based model overcame data sparsity problems and performed well in the temporally split problem setting and imbalanced classification. This performance was verified on the problem of subscription behavior as well. The exploratory analysis of different types of features and representations helped us decide which features to focus

our attention on. In the case of the New Yorker dataset, behavioral features were found to be quite effective at demographic prediction. Moreover, once these features were identified, more sophisticated techniques such as graph embeddings were used to counter the challenges associated with these features, which was data sparsity.

In Chapter 10, we also explored a unification strategy for two different sources of content, namely news and twitter. A technical challenge faced in this problem was the disparity between the two content sources. Tweets are short, contain non-standard usages of language such as emoticons, slang, spelling, and grammatical errors, and lack context due to their short length. In comparison, news articles use a formal language and provide many details surrounding the reported event. Thus, techniques that combine the representation of the two, such as topic modeling or doc2vec, do not work as well. While strategies such as aggregating the tweets by author, keywords, hashtags, or time were used to mitigate the disparity in document sizes, an entity-based strategy was ultimately found to be robust to most of the issues mentioned earlier.

We consider two scenarios of imbalanced classification. In the online prediction domain, we consider both a resampling method, which resamples the content-representation of users by generating synthetic text profiles of the minority class users. On the behavioral features side, the graph embeddings method is resilient to performance drops due to imbalance compared to the other methods.

Inspired by the effectiveness of graph embeddings in the imbalanced data case, we proposed and evaluated an algorithm to generate new features in the traditional supervised learning case with semi-structured data. This framework includes a graph construction component and learning embeddings through the constructed graph, which are used in the classifier for training and prediction. One of the biggest challenges encountered in random-walking based graph embedding methods is identifying the best-choice of random-walking strategy. While sophisticated random walks us-

ing higher-order Markov chains or teleportation can generate better embeddings, they also have a trade-off in parallelization and execution time. Ultimately, the best strategies found were the most efficient ones that were also robust to hyperparameter tuning.

Thus in this dissertation, we have focused on the problem of inferring users' characteristics and predicting them from behavior and content data while also paying particular attention to the problem of imbalance in the data distribution. While we have focused on the consumer experience, we must also consider another important implication of this work: ethics. Bias exists in the real-world and is a part of human-generated data. Unfortunately, when models are trained on this biased data, they become biased too, which is undesirable from the perspective of fairness and ethics. In this dissertation, we address the problem of bias caused by the unequal representation of users by proposing models that overcome this imbalance. Another way of using the techniques and ideas proposed in this dissertation is to identify bias in human-generated data and correct it before feeding to machine learning models.

Finally, the concern of privacy in using data to improve the products and services offered to consumers is of great importance. Care must be taken to ensure that personal information does not leak when datasets are shared across products or organizations, either for research or improving services, and the anonymity of individuals contributing to data is protected. Thus, research involving personal data must be undertaken with caution not just in how the data is shared but even the research questions asked. We hope that this dissertation provides ideas on strategies to counter ethical issues such as bias and unfairness that the community faces.

BIBLIOGRAPHY

1. Caliper analytics. URL http://www.imsglobal.org/activity/caliper.
2. URL https://www.wikidata.org/wiki/Q22686. Accessed: 2020-06-21.
3. What is the experience api? URL https://xapi.com/overview/.
4. URL https://en.wikipedia.org/wiki/Kevin_Young. Accessed: 2020-06-21.
5. Basic overview of how LTI works. URL https://www.imsglobal.org/basic-overview-how-lti-works.
6. 2014. URL https://web.archive.org/web/20141021223329/http://facultyhandbook.nd.edu:80/assets/137497/undergraduate_academic_code_final_for_7.1.2014.docx.
7. H. Abdi and L. J. Williams. Principal component analysis. *Wiley interdisciplinary reviews: computational statistics*, 2(4):433–459, 2010.
8. S. Afzal, B. Sengupta, M. Syed, N. Chawla, G. A. Ambrose, and M. Chetlur. The abc of moocs: Affect and its inter-play with behavior and cognition. In *2017 Seventh International Conference on Affective Computing and Intelligent Interaction (ACII)*, pages 279–284. IEEE, 2017.
9. J. Ah-Pine and E.-P. Soriano-Morales. A study of synthetic oversampling for twitter imbalanced sentiment analysis. 2016.
10. J. Alcalá-Fdez, A. Fernández, J. Luengo, J. Derrac, S. García, L. Sánchez, and F. Herrera. Keel data-mining software tool: data set repository, integration of algorithms and experimental analysis framework. *Journal of Multiple-Valued Logic & Soft Computing*, 17, 2011.
11. G. A. Ambrose, K. Abbott, and A. Lanski. Under the hood of a next generation digital learning environment in progress, Jul 2017. URL https://er.educause.edu/articles/2017/7/under-the-hood-of-a-next-generation-digital-learning-environment-in-progress.
12. A. Anand, K. Gorde, J. R. A. Moniz, N. Park, T. Chakraborty, and B.-T. Chu. Phishing url detection with oversampling based on text generative adversarial networks. In *2018 IEEE International Conference on Big Data (Big Data)*, pages 1168–1177. IEEE, 2018.

13. Apereo-Learning-Analytics-Initiative. Apereo learning analytics initiative - openlrw, Jul 2018. URL `https://github.com/Apereo-Learning-Analytics-Initiative/OpenLRW`.

14. K. E. Arnold, M. D. Pistilli, and K. E. Arnold. Course Signals at Purdue: Using Learning Analytics to Increase Student Success. *2nd International Conference on Learning Analytics and Knowledge*, (May):2–5, 2012. ISSN ISSN-1528-5324. doi: 10.1145/2330601.2330666.

15. S. Arora, R. Ge, and A. Moitra. Learning topic models–going beyond svd. In *2012 IEEE 53rd Annual Symposium on Foundations of Computer Science*, pages 1–10. IEEE, 2012.

16. M. Arroju, A. Hassan, and G. Farnadi. Age, gender and personality recognition using tweets in a multilingual setting. In *6th Conference and Labs of the Evaluation Forum (CLEF 2015): Experimental IR meets multilinguality, multimodality, and interaction*, pages 22–31, 2015.

17. M. Ashraf, G. A. Tahir, S. Abrar, M. Abdulaali, S. Mushtaq, and H. Mukthar. Personalized news recommendation based on multi-agent framework using social media preferences. In *2018 International Conference on Smart Computing and Electronic Enterprise (ICSCEE)*, pages 1–7. IEEE, 2018.

18. E. Auchard. Study: Men want facts, women seek personal connections on web, 2005. URL `https://www.computerworld.com/article/2560242/study--men-want-facts--women-seek-personal-connections-on-web.html`.

19. N. X. Bach, N. Do Hai, and T. M. Phuong. Personalized recommendation of stories for commenting in forum-based social media. *Information Sciences*, 352: 48–60, 2016.

20. R. S. Baker, S. K. D'Mello, M. T. Rodrigo, and A. C. Graesser. Better to be frustrated than bored: The incidence, persistence, and impact of learners' cognitive–affective states during interactions with three different computer-based learning environments. *Intl. Journal of Human-Computer Studies*, 68(4):223 – 241, 2010.

21. B. K. Baloch, S. Kumar, S. Haresh, A. Rehman, and T. Syed. Focused anchors loss: cost-sensitive learning of discriminative features for imbalanced classification. In *Asian Conference on Machine Learning*, pages 822–835, 2019.

22. M. Banerjee, M. Capozzoli, L. McSweeney, and D. Sinha. Beyond kappa: A review of interrater agreement measures. *Canadian journal of statistics*, 27(1): 3–23, 1999.

23. A. Bartoli, A. De Lorenzo, E. Medvet, and F. Tarlao. Your paper has been accepted, rejected, or whatever: Automatic Generation of Scientific Paper Reviews. In *International Conference on Availability, Reliability, and Security*. Springer, 2016.

24. S. Barua, M. M. Islam, and K. Murase. Prowsyn: Proximity weighted synthetic oversampling technique for imbalanced data set learning. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 317–328. Springer, 2013.

25. S. Barua, M. M. Islam, and K. Murase. Prowsyn: Proximity weighted synthetic oversampling technique for imbalanced data set learning. In J. Pei, V. S. Tseng, L. Cao, H. Motoda, and G. Xu, editors, *Advances in Knowledge Discovery and Data Mining*, pages 317–328, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg. ISBN 978-3-642-37456-2.

26. P. Basile and A. Caputo. Entity linking for tweets. *Encyclopedia with Semantic Computing and Robotic Intelligence*, 1(01):1630020, 2017.

27. G. E. Batista, A. L. Bazzan, M. C. Monard, et al. Balancing training data for automated annotation of keywords: a case study. In *WOB*, pages 10–18, 2003.

28. G. E. A. P. A. Batista, R. C. Prati, and M. C. Monard. A study of the behavior of several methods for balancing machine learning training data. *SIGKDD Explorations*, 6:20–29, 2004.

29. G. E. A. P. A. Batista, R. C. Prati, and M. C. Monard. A study of the behavior of several methods for balancing machine learning training data. *SIGKDD Explor. Newsl.*, 6(1):20–29, June 2004. ISSN 1931-0145. doi: 10.1145/1007730.1007735. URL `http://doi.acm.org/10.1145/1007730.1007735`.

30. C. Baziotis, N. Pelekis, and C. Doulkeridis. Datastories at semeval-2017 task 4: Deep lstm with attention for message-level and topic-based sentiment analysis. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 747–754, Vancouver, Canada, August 2017. Association for Computational Linguistics.

31. B. Berger, C. Matt, D. M. Steininger, and T. Hess. It is not just about competition with "free": Differences between content formats in consumer preferences and willingness to pay. *Journal of Management Information Systems*, 32(3): 105–128, 2015.

32. J. Bergmann and A. Sams. Flip your classroom: Reach every student in every class every day. 2012.

33. Y. Bergner, D. Kerr, and D. E. Pritchard. Methodological challenges in the analysis of MOOC data for exploring the relationship between discussion forum views and learning outcomes. *International Educational Data Mining Society*, 2015.

34. B. Bi, M. Shokouhi, M. Kosinski, and T. Graepel. Inferring the demographics of search users: Social data meets search queries. In *Proceedings of the 22nd international conference on World Wide Web*, pages 131–140. ACM, 2013.

35. J. L. Bishop, M. A. Verleger, et al. The flipped classroom: A survey of the research. In *ASEE national conference proceedings, Atlanta, GA*, volume 30, pages 1–18, 2013.

36. D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. In *Advances in neural information processing systems*, pages 601–608, 2002.

37. A. Blum, J. Lafferty, M. R. Rwebangira, and R. Reddy. Semi-supervised learning using randomized mincuts. In *Proceedings of the twenty-first international conference on Machine learning*, page 13, 2004.

38. M. Bradley and P. Lang. Affective norms for english words (anew): Instruction manual and affective ratings. *Technical Report C-1, The Center for Research in Psychophysiology, University of Florida*, 30(1):25–36, 1999.

39. M. M. Bradley and P. J. Lang. Measuring emotion: the self-assessment manikin and the semantic differential. *Journal of behavior therapy and experimental psychiatry*, 25(1):49–59, 1994.

40. C. G. Brinton, S. Buccapatnam, M. Chiang, and H. V. Poor. Mining mooc clickstreams: On the relationship between learner video-watching behavior and performance. In *ACD Conference on Knowledge Dicrovery and Data Mining (SIGKDD)*, 2015.

41. M. Broersma and T. Graham. Social media as beat: Tweets as a news source during the 2010 british and dutch elections. *journalism practice*, 6(3):403–419, 2012.

42. M. Brown, J. Dehoney, and N. Millichap. The next generation digital learning environment: A report on research, Apr 2015. URL `https://library.educause.edu/resources/2015/4/the-next-generation-digital-learning-environment-a-report-on-research`.

43. A. Bruns and J. Burgess. Researching news discussion on twitter: New methodologies. *Journalism Studies*, 13(5-6):801–814, 2012.

44. A. Bruns, T. Highfield, and R. A. Lind. Blogs, twitter, and breaking news: The produsage of citizen journalism. *Produsing theory in a digital world: The intersection of audiences and production in contemporary theory*, 80(2012):15–32, 2012.

45. E. Burnaev, P. Erofeev, and A. Papanov. Influence of resampling on accuracy of imbalanced classification. In *Eighth International Conference on Machine Vision (ICMV 2015)*, volume 9875, page 987521. International Society for Optics and Photonics, 2015.

46. H. Cai, V. W. Zheng, and K. C.-C. Chang. A comprehensive survey of graph embedding: Problems, techniques, and applications. *IEEE Transactions on Knowledge and Data Engineering*, 30(9):1616–1637, 2018.

47. B. Carstens, M. Jensen, M. Spaniel, and A. Hermansen. Vertex similarity in graphs using feature learning (2017), 2017.

48. L. Cesareo and A. Pastore. Consumers' attitude and behavior towards online music piracy and subscription-based services. *Journal of Consumer Marketing*, 31(6/7):515–525, 2014.

49. R. Chakraborty, M. Bhavsar, S. Dandapat, and J. Chandra. A network based stratification approach for summarizing relevant comment tweets of news articles. In *International Conference on Web Information Systems Engineering*, pages 33–48. Springer, 2017.

50. N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. Smote: Synthetic minority over-sampling technique. *J. Artif. Intell. Res.*, 16:321–357, 2002.

51. N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357, 2002.

52. E. Chen, Y. Lin, H. Xiong, Q. Luo, and H. Ma. Exploiting Probabilistic Topic Models to Improve Text Categorization under Class Imbalance. *Information Processing & Management*, 47(2):202–214, 2011.

53. J. Chen, H.-r. Fang, and Y. Saad. Fast approximate knn graph construction for high dimensional data via recursive lanczos bisection. *Journal of Machine Learning Research*, 10(Sep):1989–2012, 2009.

54. T. Chen and C. Guestrin. Xgboost: A scalable tree boosting system. In *KDD*, 2016.

55. J. C. Cheng. An exploratory study of emotional affordance of a massive open online course. *European Journal of Open, Distance and e-learning*, 17(1):43–55, 2014.

56. K.-H. Cheng, H.-T. Hou, and S.-Y. Wu. Exploring students' emotional responses and participation in an online peer assessment activity: A case study. *Interactive Learning Environments*, 22(3):271–287, 2014.

57. J. P. Chiu and E. Nichols. Named entity recognition with bidirectional lstm-cnns. *Transactions of the Association for Computational Linguistics*, 4:357–370, 2016.

58. S. P. Choi, S. S. Lam, K. C. Li, and B. T. Wong. Learning analytics at low cost: At-risk student prediction with clicker data and systematic proactive interventions. *Journal of Educational Technology & Society*, 21(2):273–290, 2018.

59. H. I. Chyi. Paying for what? how much? and why (not)? predictors of paying intent for multiplatform newspapers. *International Journal on Media Management*, 14(3):227–250, 2012.

60. K. Clark and C. D. Manning. Deep reinforcement learning for mention-ranking coreference models. *arXiv preprint arXiv:1609.08667*, 2016.

61. K. Clark and C. D. Manning. Improving coreference resolution by learning entity-level distributed representations. *arXiv preprint arXiv:1606.01323*, 2016.

62. D. Clow. The learning analytics cycle. In *Proceedings of the 2nd International Conference on Learning Analytics and Knowledge - LAK '12*, page 134, 2012. ISBN 9781450311113. doi: 10.1145/2330601.2330636. URL `http://dx.doi.org/doi:10.1145/3027385.3027396http://dl.acm.org/citation.cfm?doid=3027385.3027396http://dl.acm.org/citation.cfm?doid=2330601.2330636`.

63. L. Corrin and P. D. Barba. Exploring students ' interpretation of feedback delivered through learning analytics dashboards. *Proceedings of ascilite Dunedin 2014*, (November):629–633, 2014.

64. R. S. d Baker, M. M. T. Rodrigo, and U. E. Xolocotzin. The dynamics of affective transitions in simulation problem-solving environments. In *Intl. Conf. on Affective Computing and Intelligent Interaction*, pages 666–677. Springer, 2007.

65. A. Del Blanco, A. Serrano, M. Freire, I. Martinez-Ortiz, and B. Fernandez-Manjon. E-Learning standards and learning analytics. Can data collection be improved by using standard data models? In *IEEE Global Engineering Education Conference, EDUCON*, pages 1255–1261, 2013. ISBN 9781467361101. doi: 10.1109/EduCon.2013.6530268.

66. L. Derczynski, K. Bontcheva, and I. Roberts. Broad twitter corpus: A diverse named entity recognition resource. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1169–1179, 2016.

67. E. Diener, R. J. Larsen, S. Levine, and R. A. Emmons. Intensity and frequency: dimensions underlying positive and negative affect. *Journal of personality and social psychology*, 48(5):1253, 1985.

68. B. Dietz-Uhler and J. E. Hurn. Using learning analytics to predict (and improve) student success: A faculty perspective. *Journal of Interactive Online Learning*, 12(1):17–26, 2013.

69. J. Dillon, N. Bosch, M. Chetlur, N. Wanigasekara, G. A. Ambrose, B. Sengupta, and S. K. D'Mello. Student emotion, co-occurrence, and dropout in a MOOC context. In *EDM*, pages 353–357, 2016.

70. S. D'Mello, A. Graesser, et al. Monitoring affective trajectories during complex learning. In *Proceedings of the Cognitive Science Society*, volume 29, 2007.

71. S. D'Mello, A. Olney, C. Williams, and P. Hays. Gaze tutor: A gaze-reactive intelligent tutoring system. *International Journal of human-computer studies*, 70(5):377–398, 2012.

72. S. K. D'mello, S. D. Craig, A. Witherspoon, B. Mcdaniel, and A. Graesser. Automatic detection of learner's affect from conversational cues. *User modeling and user-adapted interaction*, 18(1-2):45–80, 2008.

73. Y. Dong, N. V. Chawla, and A. Swami. metapath2vec: Scalable representation learning for heterogeneous networks. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 135–144, 2017.

74. D. Dua and C. Graff. UCI machine learning repository, 2017. URL `http://archive.ics.uci.edu/ml`.

75. D. Duong, H. Tan, and S. Pham. Customer gender prediction based on e-commerce data. In *2016 Eighth International Conference on Knowledge and Systems Engineering (KSE)*, pages 91–95. IEEE, 2016.

76. C. Eksombatchai, P. Jindal, J. Z. Liu, Y. Liu, R. Sharma, C. Sugnet, M. Ulrich, and J. Leskovec. Pixie: A system for recommending 3+ billion items to 200+ million users in real-time. In *Proceedings of the 2018 World Wide Web Conference*, pages 1775–1784. International World Wide Web Conferences Steering Committee, 2018.

77. R. Ferguson. Learning analytics: drivers, developments and challenges. *International Journal of Technology Enhanced Learning*, 4(5-6):304–317, 2012.

78. R. Ferguson and D. Clow. Where is the evidence? In *Proceedings of the Seventh International Learning Analytics & Knowledge Conference on - LAK '17*, pages 56–65. ACM, 2017. ISBN 9781450348706. doi: 10.1145/3027385.3027396. URL `http://dx.doi.org/doi:10.1145/3027385.3027396http://dl.acm.org/citation.cfm?doid=3027385.3027396`.

79. A. Fernández. Smote for learning from imbalanced data: Progress and challenges, marking the 15-year anniversary. *J. Artif. Intell. Res.*, 61:863–905, 2018.

80. A. Fernández, S. del Río, N. V. Chawla, and F. Herrera. An insight into imbalanced big data classification: outcomes and challenges. *Complex & Intelligent Systems*, 3(2):105–120, 2017.

81. A. Fernández, S. Garcia, F. Herrera, and N. V. Chawla. Smote for learning from imbalanced data: progress and challenges, marking the 15-year anniversary. *Journal of artificial intelligence research*, 61:863–905, 2018.

82. C. Fink, J. Kopecky, and M. Morawski. Inferring gender from the content of tweets: A region specific example. In *Sixth International AAAI Conference on Weblogs and Social Media*, 2012.

83. J. R. Finkel, T. Grenager, and C. D. Manning. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 363–370, 2005.

84. M. Frasca and S. Bassis. Gene-disease prioritization through cost-sensitive graph-based methodologies. In *International Conference on Bioinformatics and Biomedical Engineering*, pages 739–751. Springer, 2016.

85. M. Frasca, A. Bertoni, M. Re, and G. Valentini. A neural network algorithm for semi-supervised node label learning from unbalanced data. *Neural Networks*, 43:84–98, 2013.

86. M. Frasca, A. Bertoni, and G. Valentini. An unbalance-aware network integration method for gene function prediction. In *MLSB 2013-Machine Learning for Systems Biology*, 2013.

87. M. Frasca, G. Grossi, J. Gliozzo, M. Mesiti, M. Notaro, P. Perlasca, A. Petrini, and G. Valentini. A gpu-based algorithm for fast node label learning in large and unbalanced biomolecular networks. *BMC bioinformatics*, 19(10):269, 2018.

88. J. A. Fredricks and W. McColskey. The measurement of student engagement: A comparative analysis of various methods and student self-report instruments. In *Handbook of research on student engagement*, pages 763–782. Springer, 2012.

89. J. Fu and S. Lee. Certainty-based active learning for sampling imbalanced datasets. *Neurocomputing*, 119:350–358, 2013.

90. D. Fuchs and L. S. Fuchs. Introduction to response to intervention: What, why, and how valid is it? *Reading research quarterly*, 41(1):93–99, 2006.

91. L. Gallegos, K. Lerman, A. Huang, and D. Garcia. Geography of emotion: Where in a city are people happier? In *WWW*, pages 569–574. International World Wide Web Conferences Steering Committee, 2016.

92. M. Gao, L. Chen, X. He, and A. Zhou. Bine: Bipartite network embedding. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 715–724. ACM, 2018.

93. L. S. Garavalia and M. E. Gredler. An exploratory study of academic goal setting, achievement calibration and self-regulated learning. *Journal of instructional psychology*, 29(4):221–231, 2002.

94. D. Gaševic, J. Jovanovic, A. Pardo, and S. Dawson. Detecting learning strategies with analytics: Links with self-reported measures and academic performance. *Journal of Learning Analytics*, 4(2):113–128, 2017.

95. S. Gazzah and N. E. B. Amara. New oversampling approaches based on polynomial fitting for imbalanced data sets. In *2008 The Eighth IAPR International Workshop on Document Analysis Systems*, pages 677–684, 2008.

96. A. Ghose and S. P. Han. An empirical analysis of user content generation and usage behavior on the mobile internet. *Management Science*, 57(9):1671–1691, 2011.

97. K. Goodman. First-year seminars increase persistence and retention. *First-Year Programs*, 2006.

98. M. Gori, A. Pucci, V. Roma, and I. Siena. Itemrank: A random-walk based scoring algorithm for recommender engines. In *IJCAI*, volume 7, pages 2766–2771, 2007.

99. M. Goyanes. An empirical study of factors that influence the willingness to pay for online news. *Journalism Practice*, 8(6):742–757, 2014.

100. M. Goyanes. The value of proximity: Examining the willingness to pay for online local news. *International Journal of Communication*, 9:18, 2015.

101. A. Graesser and S. K. D'Mello. Theoretical perspectives on affect and deep learning. In *New perspectives on affect and learning technologies*, pages 11–21. Springer, 2011.

102. G. Graybeal, A. Sindik, and Q. Qing. Current print subscribers more likely to pay for online. *Newspaper Research Journal*, 33(3):21, 2012.

103. A. Grover and J. Leskovec. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 855–864. ACM, 2016.

104. J.-C. Gu, Z.-H. Ling, and N. Indurkhya. A study on improving end-to-end neural coreference resolution. In *Chinese Computational Linguistics and Natural Language Processing Based on Naturally Annotated Big Data*, pages 159–169. Springer, 2018.

105. J. A. Gulla, C. Marco, A. D. Fidjestøl, J. E. Ingvaldsen, and Ö. Özgöbek. The intricacies of time in news recommendation. In *UMAP (Extended Proceedings)*, 2016.

106. W. Guo, H. Li, H. Ji, and M. Diab. Linking tweets to news: A framework to enrich short text data in social media. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 239–249, 2013.

107. G. Haixiang, L. Yijing, J. Shang, G. Mingyun, H. Yuanyue, and G. Bing. Learning from class-imbalanced data: Review of methods and applications. *Expert Systems with Applications*, 73:220–239, 2017.

108. D. K. Hatch and C. E. Garcia. Academic advising and the persistence intentions of community college students in their first weeks in college. *The Review of Higher Education*, 40(3):353–390, 2017.

109. X. He, T. Chen, M.-Y. Kan, and X. Chen. Trirank: Review-aware explainable recommendation by modeling aspects. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, pages 1661–1670. ACM, 2015.

110. J. Herbert and N. Thurman. Paid content strategies for news websites: An empirical study of british newspapers' online business models. *Journalism practice*, 1(2):208–226, 2007.

111. A. Hermida. Twittering the news: The emergence of ambient journalism. *Journalism practice*, 4(3):297–308, 2010.

112. A. Hermida, F. Fletcher, D. Korell, and D. Logan. Share, like, recommend: Decoding the social media news consumer. *Journalism studies*, 13(5-6):815–824, 2012.

113. A. E. Holton, K. H. Baek, M. Coddington, and C. Yaschur. Soliciting reciprocity: Socializing, communality, and other motivations for linking on twitter. In *International Symposium on Online Journalism, Austin, TX, April*, pages 19–20, 2013.

114. S. Hong. Online news on twitter: Newspapers' social media adoption and their online readership. *Information Economics and Policy*, 24(1):69–74, 2012.

115. X. Hong, S. Chen, and C. J. Harris. A kernel-based two-class classifier for imbalanced data sets. *IEEE Transactions on neural networks*, 18(1):28–41, 2007.

116. M. Honnibal and I. Montani. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear, 2017.

117. C.-K. Hsieh, L. Yang, H. Wei, M. Naaman, and D. Estrin. Immersive recommendation: News and event recommendations using personal digital traces. In *Proceedings of the 25th International Conference on World Wide Web*, pages 51–62, 2016.

118. J. Hu, H.-J. Zeng, H. Li, C. Niu, and Z. Chen. Demographic prediction based on user's browsing behavior. In *Proceedings of the 16th international conference on World Wide Web*, pages 151–160. ACM, 2007.

119. M. Hu, S. Liu, F. Wei, Y. Wu, J. Stasko, and K.-L. Ma. Breaking news on twitter. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 2751–2754, 2012.

120. M. Indu and K. Kavitha. Review on text summarization evaluation methods. In *2016 International Conference on Research Advances in Integrated Navigation Systems (RAINS)*, pages 1–4. IEEE, 2016.

121. T. Iwata, K. Saito, and T. Yamada. Recommendation method for extending subscription periods. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 574–579. ACM, 2006.

122. T. Jebara, J. Wang, and S.-F. Chang. Graph construction and b-matching for semi-supervised learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 441–448, 2009.

123. R. Jindal, R. Malhotra, and A. Jain. Techniques for Text Classification: Literature Review and Current Trends. *webology*, 12(2), 2015.

124. J. M. Johnson and T. M. Khoshgoftaar. Survey on deep learning with class imbalance. *Journal of Big Data*, 6(1):27, 2019.

125. E. Jones, T. Oliphant, P. Peterson, et al. SciPy: Open source scientific tools for Python, 2001. URL `http://www.scipy.org/`.

126. S. Kabbur, E.-H. Han, and G. Karypis. Content-based methods for predicting web-site demographic attributes. In *2010 IEEE International Conference on Data Mining*, pages 863–868. IEEE, 2010.

127. S. M. Keaveney and M. Parthasarathy. Customer switching behavior in online services: An exploratory study of the role of selected attitudinal, behavioral, and demographic factors. *Journal of the academy of marketing science*, 29(4): 374–390, 2001.

128. H. Khalil and M. Ebner. Moocs completion rates and possible methods to improve retention-a literature review. In *EdMedia+ innovate learning*, pages 1305–1313. Association for the Advancement of Computing in Education (AACE), 2014.

129. C. A. Kilgo, J. K. E. Sheets, and E. T. Pascarella. The link between high-impact practices and student learning: Some longitudinal evidence. *Higher Education*, 69(4):509–525, 2015.

130. R. F. Kizilcec, C. Piech, and E. Schneider. Deconstructing disengagement: analyzing learner subpopulations in massive open online courses. In *Intl. Conf. on Learning Analytics and Knowledge*, pages 170–179. ACM, 2013.

131. F. Koto. Smote-out, smote-cosine, and selected-smote: An enhancement strategy to handle imbalance in data level. *2014 International Conference on Advanced Computer Science and Information System*, pages 280–284, 2014.

132. S. Kourogi, H. Fujishiro, A. Kimura, and H. Nishikawa. Identifying attractive news headlines for social media. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, pages 1859–1862, 2015.

133. G. Kovács. smote-variants: a python implementation of 85 minority oversampling techniques. *Neurocomputing*, 366:352–354, 2019. doi: 10.1016/j.neucom.2019.06.100. (IF-2019=4.07).

134. B. Krawczyk. Learning from imbalanced data: open challenges and future directions. *Progress in Artificial Intelligence*, 5(4):221–232, 2016.

135. G. D. Kuh. What Student Affairs Professionals Need to Know About Student Engagement. *Journal of College Student Development*, 50(6):683–706, 2009. ISSN 1543-3382. doi: 10.1353/csd.0.0099. URL `http://muse.jhu.edu/content/crossref/journals/journal{_}of{_}college{_}student{_}development/v050/50.6.kuh.html`.

136. G. D. Kuh. *High-impact educational practices: what they are, who has access to them, and why they matter*. Association of American Colleges and Universities, 2008. URL `https://www.aacu.org/publications-research/publications/high-impact-educational-practices-what-they-are-who-has-access-0`.

137. N. Kumar, A. Yadandla, K. Suryamukhi, N. Ranabothu, S. Boya, and M. Singh. Arousal prediction of news articles in social media. In *International Conference on Mining Intelligence and Knowledge Exploration*, pages 308–319. Springer, 2017.

138. G. LaFree and L. Dugan. Introducing the global terrorism database. *Terrorism and political violence*, 19(2):181–204, 2007.

139. G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, and C. Dyer. Neural architectures for named entity recognition. *arXiv preprint arXiv:1603.01360*, 2016.

140. K. Lee, L. He, M. Lewis, and L. Zettlemoyer. End-to-end neural coreference resolution. *arXiv preprint arXiv:1707.07045*, 2017.

141. S.-K. Lee, S.-J. Hong, and S.-I. Yang. Oversampling for imbalanced data classification using adversarial network. In *2018 International Conference on Information and Communication Technology Convergence (ICTC)*, pages 1255–1257. IEEE, 2018.

142. W.-J. Lee, K.-J. Oh, C.-G. Lim, and H.-J. Choi. User profile extraction from twitter for personalized news recommendation. In *16th International conference on advanced communication technology*, pages 779–783. IEEE, 2014.

143. K. Leetaru and P. A. Schrodt. Gdelt: Global data on events, location, and tone, 1979–2012. In *ISA annual convention*, volume 2, pages 1–49. Citeseer, 2013.

144. J. L. Leevy, T. M. Khoshgoftaar, R. A. Bauder, and N. Seliya. A survey on addressing high-class imbalance in big data. *Journal of Big Data*, 5(1):42, 2018.

145. J. Lehmann, C. Castillo, M. Lalmas, and E. Zuckerman. Transient news crowds in social media. In *Seventh International AAAI Conference on Weblogs and Social Media*, 2013.

146. G. Lemaître, F. Nogueira, and C. K. Aridas. Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning. *Journal of Machine Learning Research*, 18(17):1–5, 2017. URL `http://jmlr.org/papers/v18/16-365`.

147. D. Leony, P. J. M. Merino, J. A. R. Valiente, A. Pardo, and C. D. Kloos. Detection and evaluation of emotions in massive open online courses. *J. UCS*, 21(5):638–655, 2015.

148. C. Li, J. Weng, Q. He, Y. Yao, A. Datta, A. Sun, and B.-S. Lee. Twiner: named entity recognition in targeted twitter stream. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*, pages 721–730, 2012.

149. F. Li, C. Yu, N. Yang, F. Xia, G. Li, and F. Kaveh-Yazdy. Iterative nearest neighborhood oversampling in semisupervised learning from imbalanced data. *The Scientific World Journal*, 2013, 2013.

150. F. Li, G. Li, N. Yang, F. Xia, and C. Yu. Label matrix normalization for semisupervised learning from imbalanced data. *New Review of Hypermedia and Multimedia*, 20(1):5–23, 2014.

151. M. Li, J. Wang, W. Tong, H. Yu, X. Ma, Y. Chen, H. Cai, and J. Han. Eknot: event knowledge from news and opinions in twitter. In *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

152. Q. Li, J. Wang, Y. P. Chen, and Z. Lin. User comments for news recommendation in forum-based social media. *Information Sciences*, 180(24):4929–4939, 2010.

153. N. Limsopatham and N. Collier. Bidirectional lstm for named entity recognition in twitter messages. 2016.

154. C. Lin, R. Xie, X. Guan, L. Li, and T. Li. Personalized news recommendation via implicit social experts. *Information Sciences*, 254:1–18, 2014.

155. C.-Y. Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81, 2004.

156. T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017.

157. C. X. Ling and V. S. Sheng. Cost-sensitive learning and the class imbalance problem, 2008.

158. Z. Liu, W. Zhang, J. Sun, H. N. Cheng, X. Peng, and S. Liu. Emotion and associated topic detection for course comments in a mooc platform. In *2016 International Conference on Educational Innovation through Technology (EITT)*, pages 15–19. IEEE, 2016.

159. T. R. Liyanagunawardena, P. Parslow, and S. Williams. Dropout: Mooc participants' perspective. 2014.

160. E. Loper and S. Bird. NLTK: The Natural Language Toolkit. In *In Proceedings of the ACL Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics. Philadelphia: Association for Computational Linguistics*, 2002.

161. M. Maier, U. V. Luxburg, and M. Hein. Influence of graph construction on graph-based clustering measures. In *Advances in neural information processing systems*, pages 1025–1032, 2009.

162. C. D. Manning, M. Surdeanu, J. Bauer, J. R. Finkel, S. Bethard, and D. McClosky. The stanford corenlp natural language processing toolkit. In *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations*, pages 55–60, 2014.

163. M. Marelli, L. Bentivogli, M. Baroni, R. Bernardi, S. Menini, and R. Zamparelli. Semeval-2014 task 1: Evaluation of compositional distributional. *SemEval-2014*, 2014.

164. A. I. Marqués, V. García, and J. S. Sánchez. On the suitability of resampling techniques for the class imbalance problem in credit scoring. *Journal of the Operational Research Society*, 64(7):1060–1070, 2013.

165. K. D. Mattingly, M. C. Rice, and Z. L. Berge. Learning analytics as a tool for closing the assessment loop in higher education. *Knowledge Management & E-Learning: An International Journal (KM&EL)*, 4(3):236–247, 2012.

166. J. G. Mazoue. The mooc model: Challenging traditional education. 2014.

167. A. McCallum and W. Li. Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. 2003.

168. R. J. L. McQuiggan, S. W. and J. C. Lester. Affective transitions in narrative-centered learning environments. *Educational Technology & Society*, 13(1):40–53, 2010.

169. N. Mehrabi, F. Morstatter, N. Saxena, K. Lerman, and A. Galstyan. A survey on bias and fairness in machine learning. *arXiv preprint arXiv:1908.09635*, 2019.

170. I. Mele, S. A. Bahrainian, and F. Crestani. Event mining and timeliness analysis from heterogeneous news streams. *Information Processing & Management*, 56 (3):969–993, 2019.

171. Y. Mi. Imbalanced classification based on active learning smote. *Research Journal of Applied Science Engineering and Technology*, 5:944–949, 2013.

172. T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

173. T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

174. P. Miller and X. Duan. Ngdle learning analytics: Gaining a 360-degree view of learning, Jan 2018. URL `https://er.educause.edu/blogs/2018/1/ngdle-learning-analytics-gaining-a-360-degree-view-of-learning`.

175. Z. Miller, B. Dickinson, and W. Hu. Gender prediction on twitter using stream algorithms with n-gram character features. *International Journal of Intelligence Science*, 2(04):143, 2012.

176. J. Misztal-Radecka. Building semantic user profile for polish web news portal. *Computer Science*, 19:307–332, 2018.

177. S. M. Mohammad. Sentiment analysis: Detecting valence, emotions, and other affectual states from text. In *Emotion measurement*, pages 201–237. Elsevier, 2016.

178. S. Moon. High-Impact Educational Practices as Promoting Student Retention and Success. pages 203–221, 2013.

179. Y. Mor, R. Ferguson, and B. Wasson. Editorial: Learning design, teacher inquiry into student learning and learning analytics: A call for action. *British Journal of Educational Technology*, 46(2):221–229, 2015. ISSN 14678535. doi: 10.1111/bjet.12273.

180. A. More. Survey of resampling techniques for improving classification performance in unbalanced datasets. *arXiv preprint arXiv:1608.06048*, 2016.

181. J. S. Morgan, C. Lampe, and M. Z. Shafiq. Is news sharing on twitter ideologically biased? In *Proceedings of the 2013 conference on Computer supported cooperative work*, pages 887–896, 2013.

182. J. Murphy and M. Roser. Internet. *Our World in Data*, 2019. https://ourworldindata.org/internet.

183. P. A. Murtaugh, L. D. Burns, and J. Schuster. Predicting the Retention of University Students. *Research in Higher Education*, 40(3):355–371, 1999. ISSN 1573-188X. doi: 10.1023/A:1018755201899.

184. M. Myllylahti. Newspaper paywalls—the hype and the reality: A study of how paid news content impacts on media corporation revenues. *Digital journalism*, 2(2):179–194, 2014.

185. M. Myllylahti. What content is worth locking behind a paywall? digital news commodification in leading australasian financial newspapers. *Digital Journalism*, 5(4):460–471, 2017.

186. A. Nigam, S. Aguinaga, and N. V. Chawla. Connecting the dots to infer followers' topical interest on twitter. In *2016 International Conference on Behavioral, Economic and Socio-cultural Computing (BESC)*, pages 1–6. IEEE, 2016.

187. A. Nigam, R. A. Johnson, D. Wang, and N. V. Chawla. Characterizing online health and wellness information consumption: A study. *Information Fusion*, 46: 33–43, 2019.

188. A. N. Nikolakopoulos and G. Karypis. Recwalk: Nearly uncoupled random walks for top-n recommendation. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, pages 150–158. ACM, 2019.

189. K. OṘegan. Emotion and e-learning. *Journal of Asynchronous Learning Networks*, 7(3):78–92, 2003.

190. J. Otterbacher. Inferring gender of movie reviewers: exploiting writing style, content and metadata. In *Proceedings of the 19th ACM international conference on Information and knowledge management*, pages 369–378. ACM, 2010.

191. K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318, 2002.

192. A. Pardo, R. Martínez-Maldonado, S. Buckingham Shum, J. Schulte, S. McIntyre, D. Gašević, J. Gao, and G. Siemens. Connecting data with student support actions in a course: A Hands-on Tutorial. In *Proceedings of the Seventh International Learning Analytics & Knowledge Conference on - LAK '17*, pages 522–523, 2017. ISBN 9781450348706. doi: 10.1145/3027385.3029441.

193. Z. A. Pardos, R. S. J. D. Baker, M. O. C. Z. San Pedro, S. M. Gowda, and S. M. Gowda. Affective states and state tests: Investigating how affect throughout the school year predicts end of year learning outcomes. In *Intl. Conf. on Learning Analytics and Knowledge*, LAK '13, pages 117–124. ACM, 2013.

194. Y. Park and I.-H. Jo. Development of the Learning Analytics Dashboard to Support Students ' Learning Performance Learning Analytics Dashboards (LADs). *Journal of Universal Computer Science*, 21(1):110–133, 2015. ISSN 0948695X (ISSN). doi: 10.3217/jucs-021-01-0110. URL `http://dspace.ewha.ac.kr/bitstream/2015.oak/230480/1/001.pdf`.

195. E. Partalidou, E. Spyromitros-Xioufis, S. Doropoulos, S. Vologiannidis, and K. I. Diamantaras. Design and implementation of an open source greek pos tagger and entity recognizer using spacy. In *2019 IEEE/WIC/ACM International Conference on Web Intelligence (WI)*, pages 337–341. IEEE, 2019.

196. M. Parthasarathy and A. Bhattacherjee. Understanding post-adoption behavior in the context of online services. *Information systems research*, 9(4):362–379, 1998.

197. S. Paulussen and R. A. Harder. Social media references in newspapers: Facebook, twitter and youtube as sources in newspaper journalism. *Journalism practice*, 8(5):542–551, 2014.

198. F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

199. R. Pekrun, T. Goetz, W. Titz, and R. P. Perry. Academic emotions in students' self-regulated learning and achievement: A program of qualitative and quantitative research. *Educational psychologist*, 37(2):91–105, 2002.

200. R. Pekrun, T. Goetz, W. Titz, and R. P. Perry. Academic emotions in students' self-regulated learning and achievement: A program of qualitative and quantitative research. *Educational psychologist*, 37(2):91–105, 2002.

201. R. Pekrun, T. Goetz, A. C. Frenzel, P. Barchfeld, and R. P. Perry. Measuring emotions in students' learning and performance: The achievement emotions questionnaire (aeq). *Contemporary educational psychology*, 36(1):36–48, 2011.

202. B. Perozzi, R. Al-Rfou, and S. Skiena. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 701–710. ACM, 2014.

203. M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*, 2018.

204. T. M. Phuong et al. Gender prediction using browsing history. In *Knowledge and Systems Engineering*, pages 271–283. Springer, 2014.

205. S. R. Porter and R. L. Swing. Understanding how first-year seminars affect persistence. *Research in Higher Education*, 47(1):89–109, 2006.

206. P. Potash, A. Romanov, and A. Rumshisky. Ghostwriter: Using an LSTM for Automatic Rap Lyric Generation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1919–1924, 2015.

207. S. Priya, R. Sequeira, J. Chandra, and S. K. Dandapat. Where should one get news updates: Twitter or reddit. *Online Social Networks and Media*, 9:17–29, 2019.

208. J. Qiu, Y. Dong, H. Ma, J. Li, K. Wang, and J. Tang. Network embedding as matrix factorization: Unifying deepwalk, line, pte, and node2vec. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, pages 459–467. ACM, 2018.

209. A. Rajadesingan, R. Zafarani, and H. Liu. Sarcasm detection on twitter: A behavioral modeling approach. In *WSDM*, pages 97–106. ACM, 2015.

210. E. Ramentol, N. Verbiest, R. Bello, Y. Caballero, C. Cornelis, and F. Herrera. Smote-frst: a new resampling method using fuzzy rough set theory. In *Uncertainty Modeling in Knowledge Engineering and Decision Making*, pages 800–805. World Scientific, 2012.

211. A. Ramesh, D. Goldwasser, B. Huang, H. Daume, and L. Getoor. Understanding mooc discussion forums using seeded LDA. In *Proceedings of the Ninth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 28–33, 2014.

212. M. M. T. Rodrigo. Dynamics of student cognitive-affective transitions during a mathematics game. *Simulation & Gaming*, 42(1):85–99, 2011.

213. P. Rodriguez, A. Ortigosa, and R. M. Carro. Extracting emotions from texts in e-learning environments. In *2012 Sixth International Conference on Complex, Intelligent, and Software Intensive Systems*, pages 887–892. IEEE, 2012.

214. N. Rout, D. Mishra, and M. K. Mallick. Handling imbalanced data: a survey. In *International Proceedings on Advances in Soft Computing, Intelligent Systems and Applications*, pages 431–443. Springer, 2018.

215. V. Salonen and H. Karjaluoto. Web personalization: the state of the art and future avenues for research and practice. *Telematics and Informatics*, 33(4): 1088–1104, 2016.

216. J. L. Santos, K. Verbert, S. Govaerts, and E. Duval. Addressing learner issues with StepUp! *Proceedings of the Third International Conference on Learning Analytics and Knowledge - LAK '13*, page 14, 2013. doi: 10.1145/2460296.2460301. URL `http://dl.acm.org/citation.cfm?doid=2460296.2460301`.

217. B. Santoso, H. Wijayanto, K. Notodiputro, and B. Sartono. Synthetic over sampling methods for handling class imbalanced problems: a review. In *IOP conference series: earth and environmental science*, volume 58, page 012031. IOP Publishing, 2017.

218. B. Settles. Biomedical named entity recognition using conditional random fields and rich feature sets. In *Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and its Applications (NLPBA/BioNLP)*, pages 107–110, 2004.

219. G. Siemens. Learning analytics: The emergence of a discipline. *American Behavioral Scientist*, 57(10):1380–1400, 2013.

220. G. Siemens and P. Long. Penetrating the fog: Analytics in learning and education. *EDUCAUSE review*, 46(5):30, 2011.

221. T. Sinha, P. Jermann, N. Li, and P. Dillenbourg. Your click decides your fate: Inferring information processing and attrition behavior from mooc video clickstream interactions. In *2014 Empirical Methods in Natural Language Processing Workshop on Modeling Large Scale Social Interaction in Massively Open Online Courses*, number EPFL-TALK-202095, 2014.

222. T. Sinha, P. Jermann, N. Li, and P. Dillenbourg. Your click decides your fate: Inferring information processing and attrition behavior from mooc video clickstream interactions. *arXiv preprint arXiv:1407.7131*, 2014.

223. A. Spitz and M. Gertz. Exploring entity-centric networks in entangled news streams. In *Companion Proceedings of the The Web Conference 2018*, pages 555–563, 2018.

224. A. Sun, E.-P. Lim, and Y. Liu. On strategies for imbalanced text classification using svm: A comparative study. *Decision Support Systems*, 48(1):191–201, 2009.

225. H. K. Swecker, M. Fifolt, and L. Searby. Academic advising and first-generation college students: A quantitative study on student retention. *NACADA Journal*, 33(1):46–53, 2013.

226. M. Syed, T. Anggara, A. Lanski, X. Duan, G. A. Ambrose, and N. V. Chawla. Integrated closed-loop learning analytics scheme in a first year experience course. In *Proceedings of the 9th international conference on learning analytics & knowledge*, pages 521–530, 2019.

227. M. Syed, M. Chetlur, S. Afzal, G. A. Ambrose, and N. V. Chawla. Implicit and explicit emotions in moocs. *International Educational Data Mining Society*, 2019.

228. M. Syed, J. Marshall, A. Nigam, and N. V. Chawla. Gender prediction through synthetic resampling of user profiles using seqgans. In *International Conference on Computational Data and Social Networks*, pages 363–370. Springer, 2019.

229. J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei. Line: Large-scale information network embedding. In *Proceedings of the 24th international conference on world wide web*, pages 1067–1077, 2015.

230. A. Tatar, P. Antoniadis, M. D. De Amorim, and S. Fdida. From popularity prediction to ranking online news. *Social Network Analysis and Mining*, 4(1): 174, 2014.

231. M. Trevisiol, L. M. Aiello, R. Schifanella, and A. Jaimes. Cold-start news recommendation with domain-dependent browse graph. In *Proceedings of the 8th ACM Conference on Recommender systems*, pages 81–88, 2014.

232. M. Tsagkias, M. De Rijke, and W. Weerkamp. Hypergeometric language models for republished article finding. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, pages 485–494, 2011.

233. E. Tutubalina and S. Nikolenko. Automated prediction of demographic information from medical user reviews. In *International Conference on Mining Intelligence and Knowledge Exploration*, pages 174–184. Springer, 2016.

234. T. Urata and A. Maeda. An entity disambiguation approach based on wikipedia for entity linking in microblogs. In *2017 6th IIAI International Congress on Advanced Applied Informatics (IIAI-AAI)*, pages 334–338. IEEE, 2017.

235. J. Vianden and P. J. Barlow. Strengthen the bond: Relationships between academic advising quality and undergraduate student loyalty. *The Journal of the National Academic Advising Association*, 35(2):15–27, 2015.

236. A. J. Viera, J. M. Garrett, et al. Understanding interobserver agreement: the kappa statistic. *Fam med*, 37(5):360–363, 2005.

237. F. Vis. Twitter as a reporting tool for breaking news: Journalists tweeting the 2011 uk riots. *Digital journalism*, 1(1):27–47, 2013.

238. J. Waitelonis and H. Sack. Named entity linking in# tweets with kea. In # *Microposts*, pages 61–63, 2016.

239. G. Wang, X. Zhang, S. Tang, H. Zheng, and B. Y. Zhao. Unsupervised clickstream clustering for user behavior analysis. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, pages 225–236. ACM, 2016.

240. J. Wang, W. Tong, H. Yu, M. Li, X. Ma, H. Cai, T. Hanratty, and J. Han. Mining multi-aspect reflection of news events in twitter: Discovery, linking and presentation. In *2015 IEEE International Conference on Data Mining*, pages 429–438. IEEE, 2015.

241. L. Wang, J. Zhan, C. Luo, Y. Zhu, Q. Yang, Y. He, W. Gao, Z. Jia, Y. Shi, S. Zhang, et al. Bigdatabench: A big data benchmark suite from internet services. In *2014 IEEE 20th International Symposium on High Performance Computer Architecture (HPCA)*, pages 488–499. IEEE, 2014.

242. A. B. Warriner, V. Kuperman, and M. Brysbaert. Norms of valence, arousal, and dominance for 13,915 english lemmas. *Behavior research methods*, 45(4): 1191–1207, 2013.

243. J. Wei, Z. Shen, N. Sundaresan, and K.-L. Ma. Visual cluster exploration of web clickstream data. In *2012 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pages 3–12. IEEE, 2012.

244. W. Wei, J. Li, L. Cao, Y. Ou, and J. Chen. Effective detection of sophisticated online banking fraud on extremely imbalanced data. *World Wide Web*, 16(4): 449–475, 2013.

245. Z. Wei and W. Gao. Gibberish, assistant, or master? using tweets linking to news for extractive single-document summarization. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1003–1006, 2015.

246. M. Wen, D. Yang, and C. Rose. Sentiment analysis in MOOC discussion forums: What does it tell us? In *EDM*. Citeseer, 2014.

247. J. Wihbey, T. D. Coleman, K. Joseph, and D. Lazer. Exploring the ideological nature of journalists' social networks on twitter and associations with news story content. *arXiv preprint arXiv:1708.06727*, 2017.

248. P. H. Winne and D. Jamieson-Noel. Exploring students' calibration of self reports about study tactics and achievement. *Contemporary Educational Psychology*, 27(4):551–572, 2002.

249. A. F. Wise. Designing pedagogical interventions to support student use of learning analytics. In *Proceedings of the fourth international conference on learning analytics and knowledge*, pages 203–211. ACM, 2014.

250. A. Wolff, Z. Zdrahal, A. Nikolov, and M. Pantucek. Improving retention: Predicting at-risk students by analysing clicking behaviour in a virtual learning environment. *LAK '13 Proceedings of the Third International Conference on Learning Analytics and Knowledge*, 2013. doi: 10.1145/2460296.2460324.

251. M. Wosnitza and S. Volet. Origin, direction and impact of emotions in social online learning. *Learning and instruction*, 15(5):449–464, 2005.

252. M. Wosnitza and S. Volet. Origin, direction and impact of emotions in social online learning. *Learning and instruction*, 15(5):449–464, 2005.

253. J. Wu, J. He, and Y. Liu. Imverde: Vertex-diminished random walk for learning network representation from imbalanced data. *arXiv preprint arXiv:1804.09222*, 2018.

254. V. Yadav and S. Bethard. A survey on recent advances in named entity recognition from deep learning models. *arXiv preprint arXiv:1910.11470*, 2019.

255. I. Yamada, H. Takeda, and Y. Takefuji. Enhancing named entity recognition in twitter messages using entity linking. In *Proceedings of the Workshop on Noisy User-generated Text*, pages 136–140, 2015.

256. D. Yang, M. Wen, I. Howley, R. Kraut, and C. Rose. Exploring the effect of confusion in discussion forums of massive open online courses. In *L@S*, pages 121–130. ACM, 2015.

257. Z. Yang, W. W. Cohen, and R. Salakhutdinov. Revisiting semi-supervised learning with graph embeddings. *arXiv preprint arXiv:1603.08861*, 2016.

258. L. Yu, W. Zhang, J. Wang, and Y. Yu. Seqgan: Sequence generative adversarial nets with policy gradient. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

259. C. Zhang and P. Zhang. Predicting gender from blog posts. *University of Massachussetts Amherst, USA*, 2010.

260. D. Zhang, J. Yin, X. Zhu, and C. Zhang. Network representation learning: A survey. *IEEE transactions on Big Data*, 2018.

261. Z.-W. Zhang, X.-Y. Jing, and T.-J. Wang. Label propagation based semi-supervised learning for software defect prediction. *Automated Software Engineering*, 24(1):47–69, 2017.

262. Z. Zheng, Y. Cai, and Y. Li. Oversampling method for imbalanced classification. *Computing and Informatics*, 34(5):1017–1037, 2016.

263. M. Zhou and P. H. Winne. Modeling academic achievement by self-reported versus traced goal orientation. *Learning and Instruction*, 22(6):413–419, 2012.

264. X. Zhu, Y. Liu, Z. Qin, and J. Li. Data Augmentation in Emotion Classification using Generative Adversarial Networks. *ArXiv*, abs/1711.00648, 2017.