Indian Institute Of Technology Delhi

Major Project Thesis

Predicting Virality and Adoption of Topics in Online Social Networks

INDIAN INSTITUTE OF TECHNOLOGY DELHI

Abstract

Integrated M.Tech. in Mathematics and Computing

Predicting Virality and Adoption of Topics in Online Social Networks

by Harvineet Singh

Online social networks play an increasingly ubiquitous role in both the discovery and dissemination of information among their users. The properties of the spread of topics on these networks exhibit large and seemingly unpredictable variation. This project aims at predicting two aspects of this spread, i.e. its scale and the individuals participating in it. In the first part, we address the question of whether virality of topics can be predicted from their early spreading patterns. We analyse the efficacy of a diverse set of features for predicting virality and establish the importance of network based features for this task.

In the second part, we look at the problem of predicting the future adopters of a topic. In contrast to previous work which relies on the link structure of the network to model the diffusion of topics, we approach this task by modelling the temporal dynamics of topic adoptions. Thus, the proposed technique does not assume any knowledge of the underlying network. For this, a representation learning based approach is proposed which embeds users in a real-valued low dimensional vector space based on their similarity in adoption behaviour. Our methodology builds on techniques initially introduced in the context of natural language processing that we use to discover hidden representations of users in social networks. We further validate the versatility of the representations by using them for the task of inferring geo-location of users. Experiments to validate the proposed methods are done on a large-scale network consisting of about 7.7 million Twitter users and tweets generated by them over a period of one month.

Acknowledgements

I wish to express my sincerest gratitude to Prof. Amitabha Tripathi, Prof. Amitabha Bagchi and Prof. Parag Singla for their support and mentorship throughout the course of the project. They have been tremendously patient in guiding me and providing me with new ideas to explore. Their timely advice and encouragement has kept me motivated. I would like to give my sincerest thanks to Prof. Amitabha Bagchi for being considerate and appreciative of the work. His ingenuity in explaining difficult concepts in a simple and elegant way is a life lesson. I am grateful to Prof. Parag Singla for involving me with this project and also for instilling in me a great interest in the field of machine learning through his courses and discussions as part of the project.

I would like to thank Siddharth Bora, who has been a constant source of inspiration and advice on all matters related to project and otherwise. His work as part of his Master's project has been instrumental in making the present study possible. The work done with him in the initial part of the project was a pleasant and enriching experience.

I would also like to thank Anirban Sen for his help with parts of the project and generously giving his time for discussing and helping me with problems related to the project.

Finally, my sincerest thanks to the staff at Department of CS&E for providing computing and laboratory facilities along with timely support in accessing these.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Online social networking services provide their users with tools for consuming, generating and sharing a large amount of information. Online platforms like *Facebook*, *Twitter*, *LinkedIn* etc. facilitate sharing of ideas, interests, activities, events and news from one user to another. The range of applications that they enable and enhance are very diverse and of tremendous importance which include facilitating social interactions, spreading of news or opinions, assisting emergency disaster response, co-ordinating political and marketing campaigns, improving product recommendation among many more. Social networks, resulting from the user interactions on these online platforms, play a crucial role in driving these applications by enabling the spreading of information or behaviour from one user to another. This can be seen as diffusion of information over the network defined by its users who are connected with each other through edges that can correspond to various interactions including social relationships between them. Here, a node which has been activated by an information or behaviour can influence its inactive neighbours to activate as well, thus resulting in the diffusion of information within the network. Due to the increase in observability of such interactions at a microscopic scale and computational techniques for processing them, studying the dynamics of information diffusion in these networks has been a focus of many recent studies.
One interesting aspect of such diffusion is the variation observed in the behaviour of different pieces of information or topics (which can be of various forms such as URL of a website, images, videos, user generated tags). While most of the topics

manage to reach only a small part of the network and quickly phase out, some topics spread rapidly and reach a large part of the network, termed as *viral*. The term *viral* is used to indicate an analogy of the spread of the topics to the rapid and infectious spread of diseases. The project aims at identifying the characteristics which influence the virality of topics and identifying distinguishing features in order to predict viral topics from their early spread.
Another aspect of diffusion is the individuals who choose to adopt a topic. We propose a method to model the adoption patterns of the users and try to predict the future adopters of a topic based on its early spread. We aim at providing a methodology that automatically discovers the features for modelling the adoption patterns instead of hand-engineering them.

1.2 Preliminaries

1.2.1 Twitter

The analyses is performed on a real world network extracted from *Twitter*, which is a micro-blogging website allowing users to post 140-character long messages or *tweets*. A tweet can contain a *hashtag* i.e. a word preceded by a # character and provides a searchable index to the tweets with the same *hashtag* in addition to and more importantly providing the twitter user with a contextual cue to the topic of discussion related to that particular tweet. This is taken as the atomic unit of information or the topic of that tweet. Thus, the terms hashtag and topic are used invariably here. A user tweeting on a particular hashtag is said to have adopted it and referred to as its adopter.
Additionally, Twitter allows a directed network to form among its users. Any user a can *follow* another user b, thus receiving tweets from b. This results in a directed network where edges go from followee to follower ($b \rightarrow a$), indicating the direction of flow of information between them. And the spread of a topic or *topic evolution graph* is taken as the subgraph induced by the users who have tweeted on that topic. This time-evolving graph is used in the analysis of the spread.

1.2.2 Dataset Description

The dataset was collected as part of the study of virality prediction problem by [Bora 2014]. It comprises of a network of around 7.7 million Twitter users which constitute the nodes in the network and their followee-follower relationships give the directed edges between the nodes. The complete set of tweets posted by these users from 27 March to 29 April, 2014 were collected. In total this consists of about 0.2 billion tweets containing about 8 million hashtags. Along with the text of the tweets, the time zone associated with the twitter account of each user, the type of tweets posted (*replies*, *mentions*, *retweets*) and the time of posting were also extracted. This allowed us to construct a detailed view of user activity on each topic in the dataset for a period of one month.

1.3 Organisation

Chapter 1 gives a brief motivation for studying information propagation in online social networks and introduces concepts related to Twitter and the online social network extracted from it for the project.
Chapter 2 describes the virality prediction task, related work done, extensions to prior work and analysis of the findings.
Chapter 3 motivates the use of representation learning, followed by description of the method used for representing users and analysis of the properties of the learned representations.
Chapter 4 discusses the related work on predicting information diffusion and the proposed method is presented along with results of the experiments.
Chapter 5 concludes the report with further extensions possible to the work.

Chapter 2

Virality Prediction

2.1 Related Work

Predicting the spread of information in online social networks has been extensively studied [Guille et al. 2013]. Many different approaches to explaining virality of topics have been taken in past, where a topic can be defined as a *meme* or a unit of transferable information. These include content-centric approach [Tsur and Rappoport 2012] i.e. using the assertion that content is a strong predictor of virality. [Romero et al. 2011] studied the differences in the spread of *hashtags* in *Twitter* belonging to different topical categories based on content. Other approaches include focusing on the topological properties of the initial spread [Romero et al. 2013]. They show that the structure of the network formed by the initial adopters of a hashtag in Twitter are reasonable predictors of its future popularity. [Ardon et al. 2013] discuss the merging of clusters of users to form giant connected component as the topic becomes popular and also introduce the analysis of conductance of the topic graph as an indicator of its popularity. Recent work [Cheng et al. 2014], [Weng et al. 2013], [Weng et al. 2014], [Ma et al. 2013] shows successes in predicting future popularity of topics. The studies investigate various temporal, content-based and network structure based features for predicting the growth of a topic in online social networks.

The present work studies the virality prediction problem as formulated in [Bora 2014] . Thus, we describe this work in more detail in Section 2.2. Further work done to extend the analysis performed in it is discussed in subsequent sections.

2.2 Existing Framework

The prior work done in formulating and analysing the problem by [Bora 2014] is described.

2.2.1 Problem Definition

A topic is said to be viral if the number of tweets on the topic exceeds the *virality threshold*. There are alternative ways of quantifying virality also. [Weng et al. 2013] use a relative measure, where topics in the top-k percentile in terms of number of tweets are considered as viral. However, the labels of topics may change with time, under this definition. As new topics appear in the top-k percentile, previously viral topics might be labelled as non-viral. Thus, making it necessary to specify a time period in which topic is said to be viral. For simplicity, an absolute threshold-based definition is used. These heuristics try to quantify the exponential growth in popularity of a topic.
The task of virality prediction is defined as correctly discriminating viral topics from non-viral ones by only observing their early spread. It is modelled as a classification problem, where features are computed based on the spread of the topic till the *prediction threshold*. The virality and prediction thresholds are decided based on the distribution of total number of tweets in the observation period. The former controls the difficulty of the prediction task and the latter limits the amount of information available for prediction.

2.2.2 Model Description

The features used in the model belong to four broad categories,

- **Evolution (E)** which capture the temporal aspect of the spread and include features such as Number of adopters, Growth rate, Number of User Mentions and Retweets.
- **Network (N)** characterises the changes in topic evolution graph using features such as Number of followers of adopters of the topic, Number of Edges, Subgraph density (ratio of edges to nodes), Number of adopters with

Name	Description
Evolution based features	
NumOfAdopters	Number of adopters who tweeted on the hashtag
NumOfRT	Number of retweets (RT) on tweets within the prediction threshold
NumOfMention	Number of user mentions (@) in tweets within the prediction threshold
TimeTakenToPredThr	Growth rate of the hashtag measured in terms of time taken to reach prediction threshold
Network based features	
HeavyUsers	Number of adopters with at least 3000 followers
NumFolAdopters	Total number of followers of adopters
NumOfEdges	Number of edges in the network spread, i.e., the subgraph induced by the set of adopters
Density	Subgraph density
SelfInitAdopters	Number of Self-initiated adopters
SelfInitAdoptersFollowers	Total follower count of Self-initiated adopters
RatioOfSingletons	Ratio of Self-initiated adopters to number of adopters
RatioOfConnectedComponents	Ratio of number of weakly connected components to number of adopters
LargestSize	Size of the largest weakly connected component
RatioSecondToFirst	Ratio of sizes of the second largest to the largest weakly connected components
Geography based features	
InfectedGeo	Number of infected geographies
RatioSelfInitComm	Fraction of Self-initiated geographies
RatioCrossGeoEdges	Fraction of edges across geographies in the induced subgraph of adopters
AdoptEntropy	Adoption Entropy measures the distribution of adopters across geographies and is defined as $-\sum_i a_i \log a_i$, where a_i is the fraction of adopters in each geography i
TweetingEntropy	Tweeting Entropy measures the distribution of tweets across geographies and is defined as $-\sum_i t_i \log t_i$, where t_i is the fraction of tweets in each geography i
IntraGeoRT	Fraction of retweets occurring between users from the same geography
IntraGeoMention	Fraction of user mentions occurring between users from the same geography
Conductance based features	
CummConductance	Conductance of the subgraph induced by the set of adopters
Conduct'_k, $k = \{20, 50, 100, 250\}$	First derivative of conductance for different values of smoothing parameter k
Conduct"	Second derivative of conductance

TABLE 2.1: Feature descriptions. All features are computed using tweets within the prediction threshold only

heavy following count (greater than 3000), Number of Self-initiated adopters (adopters with no neighbouring nodes who have already adopted) and Number of Weakly connected components in topic graph.

- **Conductance (C)** Given a graph $G = (V, E)$ and a subset of nodes $S \subseteq V$, the conductance $\phi(S)$ of the set S is defined as the ratio of the number of edges outgoing from S that are incident on nodes outside S i.e.:

$$\phi(S) = \frac{|\{(u \to v) : u \in S, v \in V \backslash S\}|}{|\{(u \to v) : u \in S, v \in V\}|} \tag{2.1}$$

  Intuitively, conductance quantifies the potential that a topic has for expanding out of the subgraph of present set of adopters. In order to capture the variations in conductance as more users tweet on the topic, first and second order derivatives of conductance are added as features.

- **Geographical (G)** properties of the spread such as Number of infected geographies, Intra and Inter geography interactions, Entropy of distribution of tweets across geographies

[Bora 2014] introduce a number of other features belonging to these categories, which are listed in Table 2.1.

2.2.3 Experiment and Results

We describe the experimental setting and the results as reported by [Bora 2014]. The features used in the prediction task were computed using prediction threshold as 1500 and topics with more than 10000 tweets (virality threshold) were labeled as viral. The resulting dataset had 2810 hashtags out of which 177 (6.3%) were viral. Table 2.2 shows the results using Random Forest [Breiman 2001] classification algorithm with 500 decision trees on various combination of features comparing them with the baseline random guess model, that randomly labels a topic as viral with probability 0.5 . For evaluation Precision, Recall, F-measure and the Area under the Precision Recall Curve (AUPRC), as described next, obtained using 10-fold cross validation on the dataset have been used.

We briefly describe the evaluation metrics used: Precision, Recall, F-measure and AUPRC. The class probabilities assigned to each test data example by the learning

Features used	**Precision**	**Recall**	**F-measure**	**AUPRC**
Random Guess	6.30	50.0	11.19	6.30
All except Conductance (E,N,G)	20.99	33.33	25.76	18.6
All features (E,N,C,G)	31.71	36.72	34.03	27.5

TABLE 2.2: Values of Evaluation metrics for virality prediction task

algorithm are thresholded to get binary output using a probability threshold or cut-off, θ. This gives the predicted labels for the examples which are compared with the corresponding actual class labels to get the number of true positives (tp), false positives (fp), true negatives (tn) and false negatives (fn), where positive refers to the virality class. Then, Precision=$\frac{tp}{tp+fp}$, Recall=$\frac{tp}{tp+fn}$, and F-measure, or *F1-score* is the harmonic mean of Precision and Recall, i.e., F-measure=$\frac{2.Precision.Recall}{Precision+Recall}$. The Precision-Recall curve is obtained by varying the value of the threshold, θ. AUPRC is calculated as the numerical approximation of area under this curve. Thus, AUPRC gives a threshold-independent measure of classifier performance.
We have discussed the problem formulation, description of the model and results of the experiments. Next, we detail the work done as part of the present project to extend the analysis of virality prediction task studied by [Bora 2014].

2.3 Class Imbalance

The highly skewed distribution of classes (only 6.3% viral hashtags) in the dataset leads to the under-representation of viral class in the learning phase of classification. One of the ways to handle this imbalance problem is to change the class distribution of the training set artificially by using sampling techniques [Hulse et al. 2007]. The test set distribution is left unchanged. Different sampling techniques which were experimented with are,

- Random Undersampling which undersamples the majority class instances
- Random Oversampling which randomly oversamples instances from the minority class
- Synthetic Minority Oversampling Technique (SMOTE) which oversamples minority class by adding artificially generated instances [Chawla et al. 2011]. In SMOTE, nearest neighbours to each minority class example are taken and new minority instances are generated by selecting a subset of the nearest

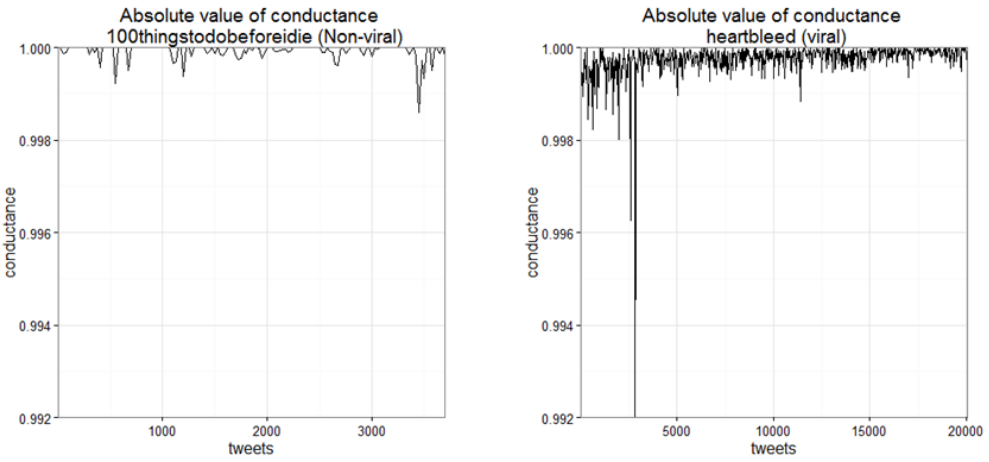


FIGURE 2.1: Variation in conductance for a viral and a non-viral hashtag

neighbours and taking a point randomly along the line joining the example and a selected neighbour.

There was an increase of 4.3% in AUPRC (the evaluation measure used to compare model accuracy) from 27.5 to 28.7 in the case of Random Undersampling when the majority class (non-viral hashtags here) was undersampled by 70% of the original majority class instances. While the AUPRC obtained with SMOTE was 28.3 .

2.4 Conductance based Features

Adding conductance-based features gave very significant increase in classification accuracy as can be seen in Table 2.2. Therefore, the dynamics of these features were studied further by looking at their values for individual hashtags.

2.4.1 Moving Window Conductance

Figure 2.1 plots the variation in conductance as the number of tweets in the observation window increases for two sample hashtags, *#100thingstodobeforeidie* (non-viral) and *#heartbleed* (viral). The conductance is computed over moving windows of length equal to 25 tweets i.e. at the arrival of each new tweet, conductance value is computed by considering the graph induced by the most recent 25 tweets which are said to lie within the moving window. This method of computing conductance differs from the earlier one in which all of the previous tweets on the

hashtag were considered instead of only the tweets in a fixed size window. It is observed that the conductance value remains very close to 1 in both the plots i.e. most of the edges in the induced graph are external, however, in case of the viral hashtag, a sharp and momentary decline in conductance is seen near 3000 tweets which corresponds to the day after the disclosure of the OpenSSL bug named *heartbleed* on April 7, 2014. The dip in the conductance value can be understood by a sudden relative increase in the number of internal edges in the induced graph because of the sharp increase in the adoptions among the followers of users who tweeted the hashtag.

2.4.2 Standard Deviation of Conductance

From Figure 2.1, we also observe the high variation in the values of conductance for the viral topics relative to the variation in case of the non-viral topic. Thus, the standard deviation of the previous conductance values was also added as a feature for classification with the hypothesis that this'll be a good discriminator for virality. Table 2.3 shows the results including standard deviation of past 100 conductance values for each hashtag as a feature. Random undersampling has also been performed on the training set to account for class imbalance problem. The

Features used	**Precision**	**Recall**	**F-measure**	**AUPRC**
Random Guess	6.30	**50.0**	11.19	6.30
Evolution only	30.00	25.42	27.52	18.5
All except Conductance	22.65	36.72	28.02	20.3
All features	**32.7**	38.98	**35.57**	**30.0**

TABLE 2.3: Values of Evaluation metrics for different models

AUPRC for the model with all the features increased by 4.5% from 28.7 to 30.0 as a result of adding standard deviation of conductance.

2.4.3 Vertex and Edge Expansion

We experimented with two other measures related to conductance which quantify the expansion property of graphs namely, *vertex* and *edge expansion*. Edge expansion for a subgraph is defined as the ratio of number of edges with only one endpoint in the subgraph by the total number of nodes in the subgraph. While, vertex expansion is the ratio of number of vertices with at least one neighbour in

the subgraph by the total number of nodes in the subgraph.
So for a hashtag, edge expansion compares the number of adopters with the number of possible exposures they create i.e. the number of follower links to in-activated users while vertex expansion compares the number of adopters with the number of inactive followers of the adopters. Using conductance, edge and vertex expansion under different combinations keeping the other feature sets unchanged did not give any improvement in the classification accuracy. It is interesting to note that out of these three measures of expansion of graphs, conductance gave the highest accuracy in predicting virality and edge expansion gave the least gain.

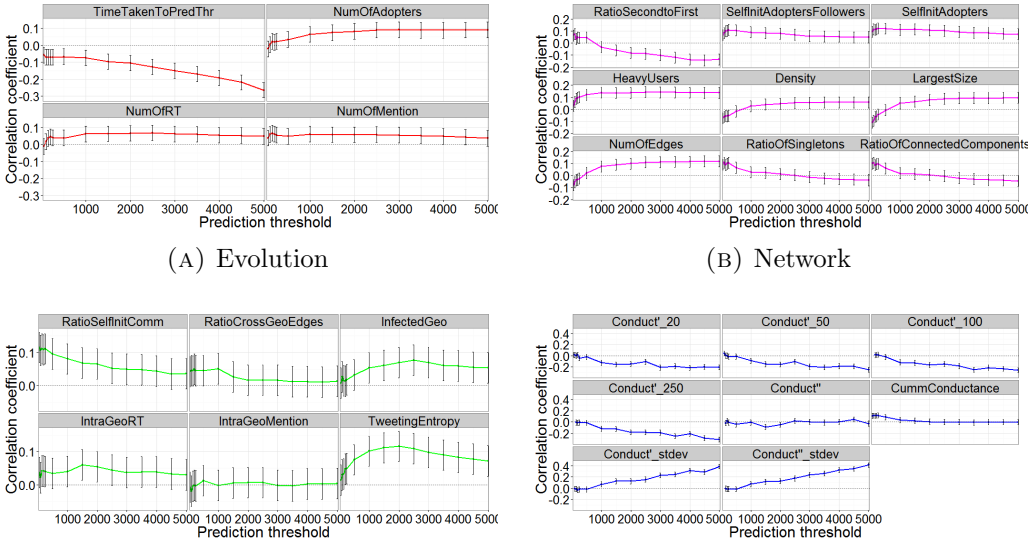
2.5 Feature Set Significance

2.5.1 Information Gain

In order to compare the contributions of each feature in classification, information gain between the feature values and the class variable was observed. For a discrete random variable Y, entropy, $H(Y) = -\sum_i P(Y = y_i)log(P(Y = y_i))$. Then, information gain between two random variables X and Y is defined as, $I(X, Y) = H(Y) - H(Y|X)$, where $H(Y|X)$ is calculated using the conditional distribution of $Y|X$. So, information gain gives a measure of the amount of information obtained about the variable Y by observing the value of variable X. Table 2.4 shows top 2 features in each feature set when they were ranked according to Information Gain.

Feature	**Set**	**Info. Gain**	**Feature**	**Set**	**Info. Gain**
1. Growth Rate	E	0.02424	1. No. Of Infected Geographies	G	0.00938
2. No. of Adopters	E	0.00979	2. Fraction of Self Initiated Geographies	G	0.0001
1. Number of Edges	N	0.0099	1. 1st Derivative of Conductance (k=50)	C	0.04527
2. No. of adopters with Heavy Following	N	0.00831	2. Stdev of 1st Derivative of Conductance	C	0.03562

TABLE 2.4: Top Ranking Features Based on Information Gain Criteria



(A) Evolution (B) Network

(C) Geography (D) Conductance

FIGURE 2.2: Correlation coefficient plots for each feature separated by the feature sets. The vertical bars represent 95% confidence intervals for the correlation coefficient estimates.

2.5.2 Correlation Analysis

Another measure of efficacy of the features in predicting virality is the correlation coefficient with the dependent variable. Since the dependent variable here is discrete-valued (viral or non-viral), total number of tweets on the topic is taken as the dependent variable. Spearman's rank correlation coefficient between the feature values and the total number of tweets on the topic is calculated. Figure 2.2 plots the variation in correlation coefficient as the number of tweets over which the features are computed increase, for all the hashtags with at least 5000 tweets. This gives us the change in importance of the features as more of the hashtag spread is observed. Some observations from these plots include,

"Time taken to prediction threshold" (TimePredThr) is negatively correlated with virality, i.e. viral topics reach the prediction threshold in short amount of time and the correlation increases (in magnitude) as number of tweets observed increases. "Number of adopters with heavy following" (HeavyUsers) is positively correlated with virality. The values for standard deviation of first and second derivative of conductance (Conduct'_stdev, Conduct"_stdev) is also increasing as the number of tweets observed increases.

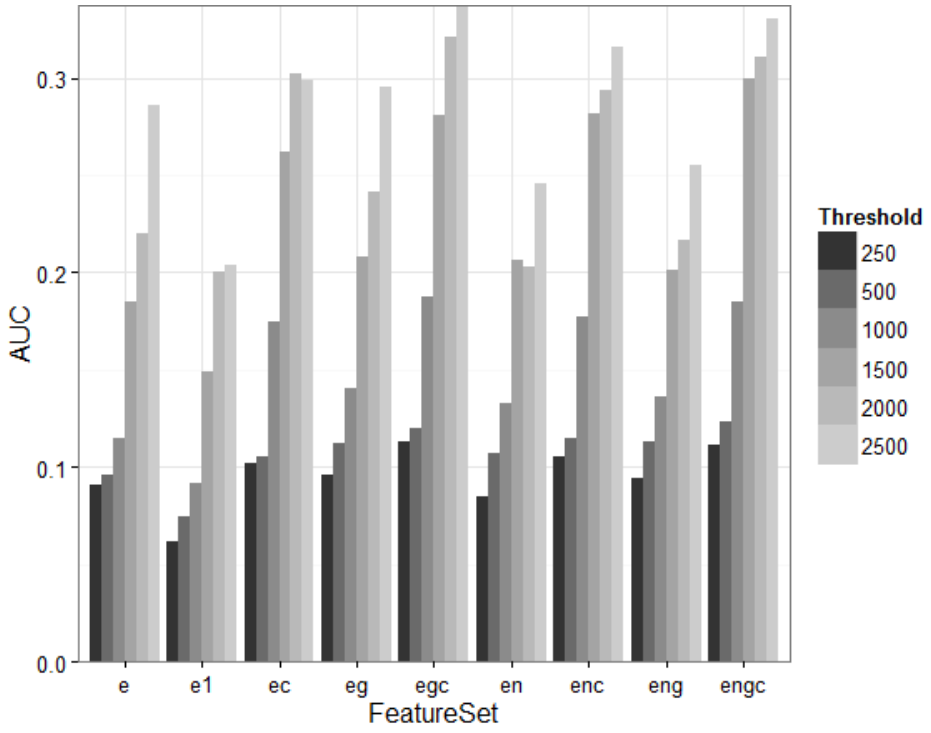


FIGURE 2.3: Performance of different feature sets at varying prediction threshold

2.6 Effect of Prediction Threshold

The prediction accuracy using different combinations of feature sets was computed at varying prediction thresholds in order to first, compare their added utility in predicting virality and second, to see how this contribution changes across different thresholds. In Figure 2.3, we observe that the AUPRC increases almost linearly from prediction threshold 250 to 1000 and from 1500 to 2500, and there is a steep increase from 1000 to 1500 across all feature set combinations. This sharp rise in AUPRC also justifies the choice of prediction threshold of 1500. It helps in balancing the trade off between obtaining sufficient amount of information to make predictions without making it necessary to observe a large part of the hashtag spread. Another observation from the figure is the considerable difference in AUPRC of the feature sets containing conductance features (EC, EGC, ENC, EGNC) as compared to the ones without conductance for prediction thresholds greater than 1000 and the absence thereof for thresholds less than 1000, suggesting that the efficacy of conductance features in predicting virality starts to be prominent around prediction threshold value of 1000.

2.7 Content based Features

The features that have been considered so far does not take into account the content of the tweets containing the hashtags. This section describes the efforts to extract content-related information from the hashtag text that can be used to characterise virality. The content-related information considered in prior work include extracting characteristics of hashtags such as length, number of word segments present, named entities present, its location within the tweet, aggregated sentiment of the tweets, among others [Tsur and Rappoport 2012]. We look at two content based properties, namely, sentiment polarity of tweets on the hashtag and the topical category of the hashtag.

2.7.1 Sentiment Classification

We focus on the dynamics of the sentiment expressed in tweets in contrast to only considering the aggregate sentiment over tweets. The hypothesis being that hashtags with more subjective content and changes in the polarity of tweets involved will be more likely to go viral as opposed to hashtags having no opinion associated with them.

For classifying tweets with positive, neutral or negative sentiment values, a twitter-specific sentiment classification service [1] was used, which is trained on tweets containing emoticons to provide training labels [Go et al. 2009]. The tweets are pre-processed to replace urls, mentions and elongated words with their normalised forms. A bag-of-words model containing both unigram and bigram features with their Part-of-Speech tags is used. For validation, they use a hand-annotated test set containing 177 negative and 182 positive tweets. The test set accuracy for the Maximum Entropy classifier is around 83% on positive and negative sentiment classification. We used the classifier trained by [Go et al. 2009] as the accuracy is sufficiently high for tweets.

The proportion of tweets in each class classified with different sentiment labels are: for viral class - Positive 18%, Neutral 79%, Negative 3% and for non-viral class - Positive 23%, Neutral 74%, Negative 3%. The proportion of positive sentiment tweets was used as a feature for virality prediction. By plotting the sentiment of tweets with time, it was observed that the sentiment of viral tweets oscillated

[1] www.sentiment140.com

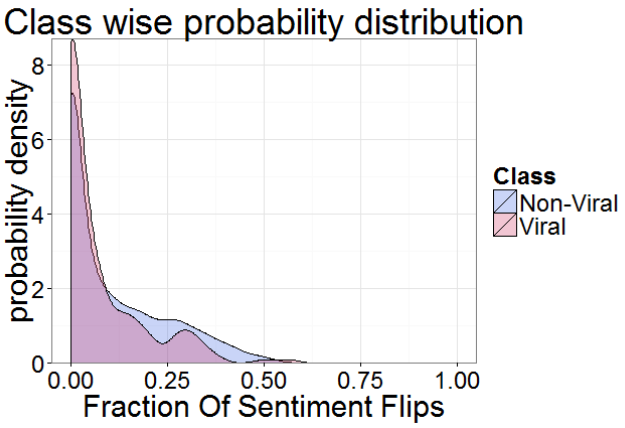


FIGURE 2.4: Class-wise distribution of fraction of flips in sentiment

among sentiment classes with time. In order to quantify this behaviour, a moving window of 50 consecutive tweets was considered and overall sentiment was calculated using majority vote of sentiment in the window (i.e. assigning the sentiment with most number of tweets). Ratio of number of flips using this aggregated sentiment to the number of 50-tweet windows considered was added as a feature for classification. Class-wise distribution of this feature is plotted in Figure 2.4. We observe that there is a significant overlap in the probability density plots, suggesting that the feature does not help in distinguishing between the two classes. Consequently, adding this feature didn't give any improvement in the classification accuracy.

2.7.2 Semantic Clustering using Word Representations

We tried to cluster the hashtags based on their topical categories like sports, politics, technology, etc. with the hypothesis that intrinsic virality of topics varies significantly among these broad categories. This was motivated by the findings of [Romero et al. 2011] that hashtags belonging to different topical categories differ in the way they spread. However, obtaining these topic-based clusters is challenging. Manually labelling each hashtag with the corresponding category with the help of human annotators is a laborious task and infeasible for such a large number of hashtags. One of the unsupervised approaches to extract topics from a collection of tweets is using topic modelling with *Latent Dirichlet allocation* (LDA) [Mehrotra et al. 2013]. However, the documents that we can use for topic modelling only include the set of tweets on the hashtag before the prediction threshold,

which limits the amount of information available for extracting topics. We focused on an alternative approach of representing words in the form of real-valued vectors referred to as *word vectors*. These can be trained on a large text corpus which can be disjoint from hashtag tweets. The resulting word vectors can be used for the task of classifying hashtags into topical categories and subsequently use this for virality prediction.

For mapping each word in our corpus into a continuous vector space we used the method proposed in [Mikolov et al. 2013a], [Mikolov et al. 2013b] (described in Section 3.3.1) using *word2vec*[2] toolbox for training word vectors on a large dataset. The training was done on a collection of about 35 million tweets collected in Nov 2013 along with the tweets from the present dataset tweeted before the prediction threshold. Basic pre-processing such as removing user mentions, URLs, special characters, elongated words and emoticons was done followed by tokenisation using Stanford Tokenizer [3] and finally converting the text to lowercase. The words were projected onto a 200-dimensional vector space. As an example of the efficacy of the word vectors learned, some sample words along with their most similar words from the corpus according to cosine similarity measure computed using word vectors are shown in Table 2.5.

Example words	**Most Similar Words**
heartbleed	openssl, cve
sfbatkid	batkid, rescue, makeawish
bjp	aap, Modi, Delhi
snowden	nsa

TABLE 2.5: Most similar words

The learning of the word vectors make use of the assumption that words appearing in same context in the text have similar meaning thus should be represented by similar vectors in the vector space. Word vectors give distributed representations of the words which capture their syntactic and semantic relationships. The hypothesis is that they can be used to segment semantically related hashtags based on the similarity of the words occurring in their tweets.

We obtained vector representations for each hashtag using the following approach. Firstly the word vectors were clustered using k-means clustering to get groups of semantically-related words. The number of clusters was set to 1000. Then the vector representing each hashtag is given by the number of words corresponding to

[2]https://code.google.com/p/word2vec/
[3]http://nlp.stanford.edu/software/tokenizer.shtml

the tweets of the hashtag that lie in each cluster. Thus a 1000-dimensional hashtag vector is obtained. This can be seen as a *bag-of-clusters* approach for representing a document (hashtag with all its tweets) instead of the standard *bag-of-words* approach. Hashtag vectors are further clustered into 10 clusters by using k-means clustering again, thus, segmenting hashtags into 10 groups. On closer inspection of the segments, it was found that the hashtags corresponding to the same language (Spanish, Italian) or same sports, events, etc. were clustered together. However, within a cluster, hashtags seemingly belonging to different themes or categories were also present. Two features were added based on this clustering, namely, the cluster number the hashtag belongs to and the size of that cluster. There was no improvement in classification accuracy (AUPRC decreased to 29.6) with the use of these features.

Thus, addition of content-based features to the existing feature set of network and geography based features did not give improvement on the predictive ability. However, using variation in conductance values and correcting for the class imbalance provided gains in prediction performance. Using these resulted in an increase of 9% in AUPRC compared to previous work [Bora 2014]. We also analysed the significance of different characteristics of topic spread in predicting virality and found that conductance related features consistently outperformed other features in their significance for the prediction task.

Chapter 3

Representation Learning in Online Social Networks

The approach that we use for predicting diffusion of information is based on representation learning. Thus, we first give an introduction of the same and give a motivation for its use in social network analysis. This is followed by the discussion on the prediction task in Chapter 4.

3.1 Motivation

The approach taken to model virality prediction problem was to look for factors that might be useful in predicting virality and extracting features or representations that aim to capture the discriminatory information presented by these factors. The hand-crafted features thus obtained were then used with standard set of classification methods. There are many potential shortcomings of the approach. The feature extraction step generally involves excessive trail-and-error. Also, since each of the features is considered in isolation, the approach becomes increasingly complex when multiple explanatory factors have to be considered. The set of features extracted for a particular tasks may not also generalise to other tasks of interest.

An alternate approach, which aims at overcoming some of these problems, is to learn good representations of the data by using supervised or unsupervised tasks defined over the data, thus, making the process of identifying features a part of

the learning algorithm itself instead of hand-crafting them. *Representation learning* involves a set of methods that learn representations of the raw data that can be effectively leveraged by standard machine learning algorithms for prediction tasks [Bengio et al. 2013]. An example of such a method is Principal Component Analysis (PCA) which learns representations of the input data points in a lower dimensional space that explain maximum amount of variability observed in the input. This approach of learning representations or features from raw data instead of hand-crafting them has been used successfully in recent work in the domains of computer vision, speech recognition and natural language processing, particularly with the use of multilayer neural networks [LeCun et al. 2015].
We aim to explore the use of such methods to learn representations of nodes in the social network. The data from social networks is characterised by the presence of different types of entities (like users, content), different attributes associated with them (like demography of users, their friendship information, topics discussed). For a given task, taking into account the complex interactions present among these entities to extract features manually can be a difficult task. We hypothesise that the representation learning approach can capture these complex interactions and leverage the rich information present in social networks which can be difficult to capture otherwise or may involve a lot of trial and error to extract an informative set of features for the given task.

3.2 Related work

There has been much recent interest in learning node representations in social networks. These representations are *learned* to capture the observed interactions among nodes, e.g. their link structure. [L. Tang et al. 2009] extract top-k eigenvectors of the modularity matrix of the network graph to get latent features for users and utilise them as features for classification tasks. Another work introduces the existing representation learning methods used in natural language processing for learning representations in social networks [Perozzi et al. 2014]. They use multiple short-truncated random walks starting from nodes in the network to generate sequences of nodes. These can be considered analogous to sentences in text. Then, *Skip-gram model* [Mikolov et al. 2013b] is used on the resulting corpus of sentences to represent each node as a real-valued vector. The learned representations capture meaningful information about the network structure as demonstrated using node

classification tasks. [J. Tang et al. 2014] propose a scalable embedding method with an objective function that explicitly encodes the *first-order* and *second-order proximity* information of users. The first-order proximity between a pair of nodes refers to information on their direct connectivity (i.e. whether they are connected or not and edge-weights in case of weighted networks), whereas second-order proximity refers to similarity in first-order proximity between nodes. An optimisation criteria is proposed that preserves both kinds of information of the nodes, while representing them in a low-dimensional space.

3.3 Learning User Representations

We used Skip-gram model to learn user representations, as proposed in [Perozzi et al. 2014]. In contrast to using link structure, we utilised topic adoption sequences which provide richer user interaction information and allowed us to model information diffusion. We discuss the Skip-gram model and then describe our methodology in more detail.

3.3.1 Skip-gram model

Skip-gram model proposed in ([Mikolov et al. 2013a],[Mikolov et al. 2013b]) is a neural network based model for learning *distributed* vector representations of words. Here, distributed means that there exists a many-to-many mapping between the dimensions of the vectors and the properties of words which the representation tries to encode. This is different from the traditional approach of representing words as *1-of-N* vectors which encodes a word's position in the vocabulary as 1 and rest of the values are 0. Thus, this encoding scheme doesn't explicitly capture the similarity between words.

Skip-gram model defines an objective function that trains the word vectors to predict their context, where context is defined as the words occurring within some fixed distance of the given word in the training corpus. Given a word u and a word c in its context, the conditional probability $p(c|u)$ is given by the *softmax function*,

$$p(c|u) = \frac{exp\ (v'_c.v_u)}{\sum_{w \in V} exp\ (v'_w.v_u)} \tag{3.1}$$

where v_w and v'_w are vector representations of word and context respectively which are parameters of the model, V is the set of words in the training corpus.
The training objective is to maximise the sum of log probabilities of all word-context pairs in the corpus:

$$\sum_{(u,v)\in C} log\ p(v|u) \tag{3.2}$$

where C is the set of all word-context pairs.
[Mikolov et al. 2013a],[Mikolov et al. 2013b] describe various techniques for efficient optimisation of this objective. The method is highly-scalable enabling word vectors to be learned from large datasets containing billions of words. As a result of the training objective chosen, words occurring in similar context are represented by similar vectors. The word vectors thus obtained have been shown to capture semantic and syntactic relationships among words.

3.3.2 Methodology

We used hashtags tweeted by the users to extract context for them. Users tweeting on the same hashtags indicate a measure of similarity among them. This similarity can be exploited for embedding users. We first describe the procedure for extracting contexts from users' tweets on hashtags.
For a hashtag h, the adoption sequence S_h is the time-ordered sequence of users' tweets on h. For each tweet in S_h and the corresponding user u, the context for u is given by the users in S_h who tweeted within a time period τ from $t(u)$. Here, $t(u)$ is the timestamp of the tweet by u. If a user tweets multiple times on a hashtag, multiple such contexts are extracted. Repeating this for all hashtags gives user-context pairs for all users. This definition of context tries to capture the users who have similar adoption patterns, i.e. similar in the topics they tweet on and the time at which they tweet.
If the tweets on a hashtag are tweeted very close by in time, this will result in a large number of contexts. One way of limiting the contexts is by varying τ for each tweet. We instead used a *path sampling* procedure. This converts S_h into a directed graph G_h, where the nodes are the tweets of S_h and an edge exists from a to b if $t(b) - t(a) \leq \tau$. Starting from a node in G_h, the next node in the path is uniformly sampled from its neighbours. This is done repeatedly until path of length γ is obtained (this is same as using a fixed-length random walk). All the nodes in the path are considered as contexts for each other. A single path

is sampled for each node in G_h. This sampling procedure limits the number of contexts for a user. This also has the advantage of capturing users tweeting at time differences larger than τ as context.
Skip-gram model is then used to learn d-dimensional user vector representations from the extracted contexts. For training, *word2vec* toolkit[1] was used. The time difference for an edge τ was set to 1 hour and path length γ was set to 10. Parameters for word2vec used were: vector dimension d 100, Skip-gram with Hierarchical Softmax, context window length 10, sub-sampling threshold 10^{-4}, training iterations 20, default values for rest of the parameters were used. User vectors were trained using tweets on 2,893,849 hashtags (80% of the total) in the dataset. User vectors for 2,574,807 users were obtained from these tweets. These were normalised to unit length post training.

3.4 Evaluation of User Representations

To evaluate the versatility of learned representations, we test them on the task of predicting geo-location of users.

3.4.1 Geo-location Inference

The time zone information of the Twitter profile of the user was taken as its geo-location. This is relatively coarse level of geographical information. While geo-tagged tweets enable finer geographical information, these were not available for most of the users. The prediction task was formulated as that of multi-class classification (number of time zones were 141). User representations were used as features for classification and the time zone of each user was taken as the target variable.

Experiment

We used *one-vs-rest* logistic regression model implemented in LibLinear toolkit [Fan et al. 2008] for classification. Users were randomly split into train and test sets with varying sizes of the training set. Overall accuracy on the test set (ratio

[1] https://code.google.com/p/word2vec/

of correctly predicted examples to the number of examples) was taken as the evaluation metric. Results were computed on a random sample of 1 million users with known geography.

For comparison, a network-based baseline was considered. This method infers the geography of a user from the most frequent geography among its neighbours [Davis et al. 2011]. The neighbours are considered as the users followed by the given user, i.e. its "friends" in Twitter terms. Ties are broken arbitrarily. If none of the neighbours' location is known for a user, then the most-frequent location in the training set is predicted. Majority guess baseline predicts the most frequently seen location in the training set for all users.

Model/% of training data (Accuracy %)	**1%**	**5%**	**10%**
Majority guess baseline	19.64	19.67	19.61
Network-based baseline (Friends)	25.40	36.58	40.52
User vectors	38.76	40.30	40.58

TABLE 3.1: Geo-location inference results

Results

Table 3.1 details the results of the experiment when the size of training set used is varied from 1% to 10%. The difference in performance is not significant when location of high percentage of users is known. However, user vector based method out-performs the baselines when less information is available. Even though the representations are not trained for this task explicitly but the good prediction performance indicates the generality of these representations. The classifier is able to generalise well from limited information of the class labels of points in the feature space. This could be because the structure of feature space is such that the users with same geography are embedded near each other, as seen in 3.1. The figure plots two-dimensional representation of user vectors, obtained using a dimensionality reduction technique called *t-SNE* [Van der Maaten et al. 2008], for a sample of users. It can be observed that users with same geography are clustered together.

While the tweets and their timestamps are readily available, the geo-location information may not be. Thus, the learned features can be used to predict geography of the users from the geographical information of only a small number of users.

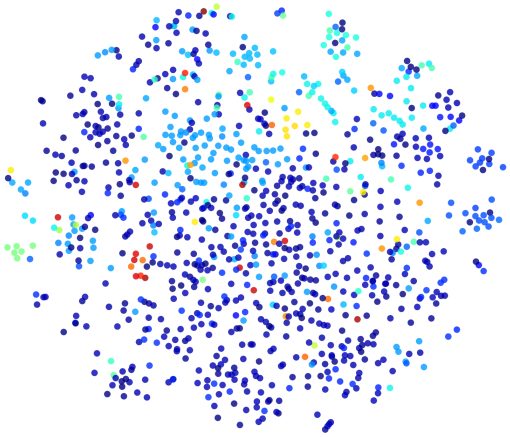


FIGURE 3.1: Visualisation of user vectors where colour represents geography of the user. 1000 randomly sampled users are plotted.

3.4.2 Properties of Embeddings

We now look at the properties of users that are preserved by these embeddings. In order to do this, we compare the neighbours of users in the vector space with their neighbours in the followee-follower network. The network neighbourhood exhibits high degree of similarity among users as the links are formed on the basis of friendships, shared interests, etc. (a phenomenon referred to as *homophily* i.e., tendency of users to form links with similar users). On the other hand, the embeddings are obtained solely on the basis of topic adoptions without considering the network structure. Therefore, we test for the similarity of the two different kinds of neighbourhoods.
In particular, we look at the following questions,

1. How similar are these neighbourhoods w.r.t. the users contained in them?
   For a particular user, we first obtain the set of its followers and the set of its k-nearest neighbours (where, k is taken to be same as number of followers) in the vector space. These two sets are then compared using Jaccard similarity index (given as cardinality of intersection of the sets divided by cardinality of union of the two). This gave a low similarity index of about 0.01 (averaged over 1000 randomly sampled users). Thus, users in two neighbourhoods are different.

2. For a user, whether its network neighbours are more likely to co-adopt topics than its neighbours in vector space?

   Firstly we define the likelihood of co-adoption for a user and its neighbourhood as follows: total number of times the user and any of its neighbour adopted the same topic divided by the total possible number of such co-adoptions, which is, number of neighbours multiplied by number of topics adopted by the user.

$$p_u = \frac{\sum_{w \in N(u)} \sum_{t \in T(u)} I\,(w \ adopts \ t)}{|N(u)|.|T(u)|} \tag{3.3}$$

   where, p_u is the likelihood of co-adoption of user u, $N(u)$ is the set of its neighbours, $T(u)$ is the set of topics adopted by u and I is the indicator function.

   We compute this measure for both network and vector space neighbours for a random sample of 10000 users. On comparison, the vector space neighbours had higher likelihood of co-adoption than network neighbours on average (0.053 and 0.0359 respectively). The same can also be observed in Figure 3.2 from the skew of the points. This also gives us an intuition into how the

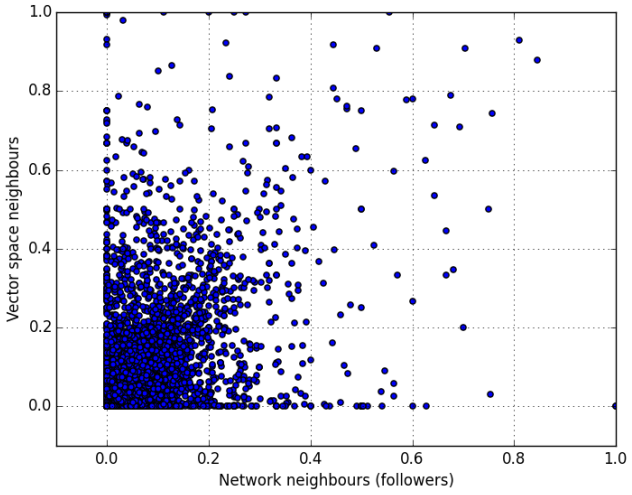


FIGURE 3.2: Scatter plot of the likelihood of co-adoption of users with different neighbourhoods. The distribution is skewed towards the upper side of the 1:1 line.

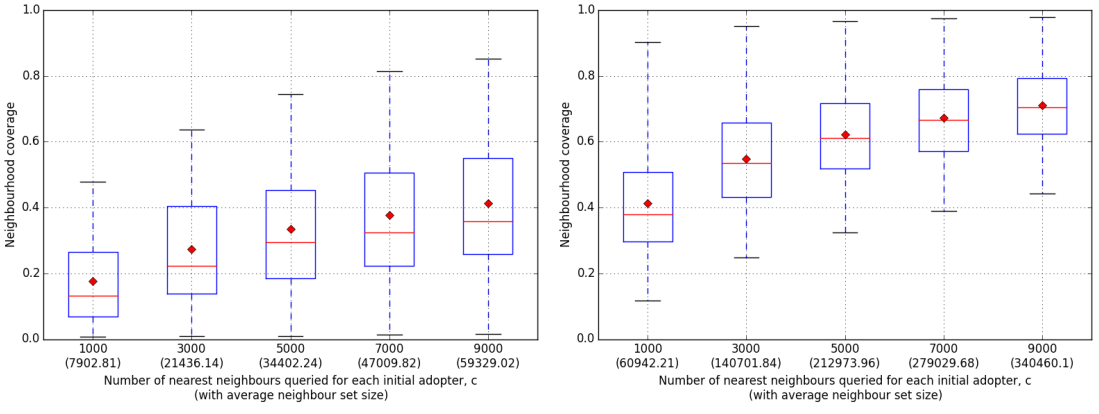
   embeddings can be used in the adopter prediction task. Instead of looking at the network neighbours of the initial adopters of a topic in order to predict

its future adopters, we can work with their vector space neighbours, as these have higher likelihood of co-adoption.

Neighbourhood Topic Coverage

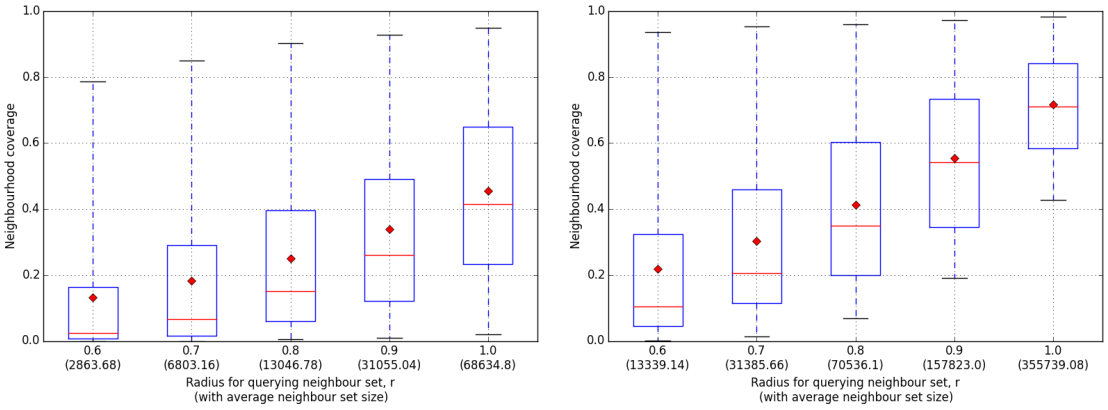
We evaluate the user representations in their ability to embed adopters of the same topics nearby each other.

For a hashtag, we consider its first n adopters and taking their vector representations query c-nearest neighbours for each of them. This collection of neighbouring users in the vector space is termed as *neighbour set* of the initial adopters. Here, Euclidean distance is used as a dis-similarity measure for finding nearest neighbours. *Neighbourhood coverage* is computed as the fraction of adopters of the hashtag present in the neighbour set to the total number of adopters. We consider 50 randomly sampled hashtags from the set of hashtags with at least 500 adopters and plot the neighbourhood coverage for $n = 10$ and $n = 100$ with different values of c (Figure 3.3). Average size of neighbour sets in each case is also given. The box plot shows mean (diamond-shaped), median, 25^{th} and 75^{th} percentiles, minimum and maximum of coverage values for each case. We note that the total number of users from which the neighbour sets are queried are 2,574,807 whereas average number of adopters of a hashtag are 2,412. So, the coverage achieved using user representations is significant.



(A) (B)

FIGURE 3.3: Neighbourhood coverage with varying c, Average over 50 topics



(A) (B)

FIGURE 3.4: Neighbourhood coverage with varying r, Average over 50 topics

Radius and k-NN Search

Similar to querying c-nearest neighbours, neighbour set can also be queried by using radius-based queries, i.e., including all users within a particular distance of the initial adopters. Figure 3.4 plots neighbourhood coverage for $n = 10$ and $n = 100$ with different values of r. Since the user vectors are normalised to unit length, r can vary between 0 and 2.

Compared to the c-nearest neighbour plots, variability is higher which is a result of the uneven spatial distribution of points, e.g. if query points are in a dense region this results in a larger neighbour set and hence larger coverage.

Chapter 4

Adopter Prediction

4.1 Related work

Models for predicting information diffusion has been extensively studied. Two widely used predictive models are *Independent Cascade model* (IC) and *Linear Threshold model* (LT) [Kempe et al. 2003]. IC model is a probabilistic model which assigns a diffusion probability for each edge. Starting from a seed set of active (or adopted) nodes, the diffusion process proceeds as follows: each newly activated node has a chance to activate its currently inactivate neighbours with the probability defined on the edge between the two. This process continues till there are no new activations. LT model in contrast is a deterministic model. It assumes an influence threshold for each node and an influence weight for each edge. It also proceeds in discrete steps, where an inactive node adopts when the sum of influence weights of its activated adopters exceeds its influence threshold. Typically, the parameters of these models such as diffusion probabilities for each edge are assumed to be known. There is also work on inferring these parameters from the historical data on diffusion of topics [Goyal et al. 2010].
However, these models assume the knowledge of the network underlying the diffusion. The diffusion network might not always be known or observable. Moreover, it is also highly dynamic. The structure of the network can change as users add and/or delete edges in response to various factors including exposure to content shared by others [Myers et al. 2014]. On the other hand, the timing information of topic adoptions is readily available and can be used to model the diffusion of topics instead of relying on the link structure.

[Bourigault et al. 2014] propose an approach that does not depend on the knowledge of the network, but instead uses the timestamps of adoption from the observed cascades (sequence of users adopting a topic). The diffusion of information is modelled using a *heat diffusion kernel*, which gives the adoption score of a user for a cascade at a given time. The aim is to learn the parameters of the kernel that maintains the temporal ordering of the observed cascades. The learned parameters are then used to predict future adopters of a cascade given its source user.

4.2 Adopter Prediction Task

Given the initial set of adopters of a topic, the task is to predict it's subsequent adopters.

4.2.1 Methodology

Given the first n adopters of a topic, the model predicts k users most likely to adopt it in future. The task can be seen as that of information retrieval, where the *query* consists of the set of first n adopters S and the *relevant documents* to be retrieved are the future adopters of the topic. For this retrieval, we need a measure to rank candidates according to their relevance to the query. We use the vector representations learned in Section 3.3 for ranking candidates. Here, candidates are taken as all the users except the initial adopters.
The ranking criteria uses a score assigned to each candidate user c which is based on the Euclidean distance of c from each user in S, measured using their corresponding vectors. We consider different ways of combining these distances:

- **Min:** $score(c, S) = \min_{a \in S} d(c, a)$, i.e. candidates are ranked in increasing order of their minimum distance to S. An implementation detail: For getting the top-k ranked list according to this criteria, k-nearest neighbours were queried for each user in S using k-d tree for efficient indexing and these lists were then merged. This provided a speed-up of about 10x compared to brute-force search, despite the high-dimensional data points.
- **Average:** $score(c, S) = \frac{1}{|S|} \sum_{a \in S} d(c, a)$, i.e. candidates are ranked in increasing order of their average distance to S. This criteria involves computing

average distance from S for each point in dataset, which is an expensive operation (given the number of points are close to 2.5 million). As an alternative to this, a candidate set is queried first by combining the k-nearest neighbours from each of the initial adopters in S. Then the candidates are ranked based on their average distance from S. Thus, this procedure amounts to finding the most similar users of the current adopters in S and ranking them based on their average distance from S to extract top-k users.

4.2.2 Experiments

The total number of topics (hashtags) in dataset are 3,617,312. For training user representations, 80% of topics and their adoption sequences (time-ordered list of users tweeting the hashtag) were used and the rest 20% topics were held-out for testing. User vectors for 2,574,807 users were obtained from the train set. Vectors were normalised to have unit length, after training. For evaluation, we considered 100 randomly sampled topics from the held-out set which have at least 500 adopters. In case if a user adopts (tweets) a topic (hashtag) multiple times then only the first adoption is taken.
For ranking candidates, we used the average based score as it performed better than the other in the experiments. Table 4.1 contains some basic statistics of the test set. Two baseline methods were considered for comparison,

Number of hashtags	2,893,849
Hashtags with $\geq$ 500 tweets	3859
Avg. no. of adopters in hashtags with $\geq$ 500 tweets	2120

TABLE 4.1: Test dataset statistics

Frequency Rank returns users in decreasing order of their frequency of adoption as observed in training topics. Thus, it predicts the same users irrespective of the topic.

Exposure Rank ranks users according to the number of possible exposures to the topic from their neighbours, which is given by number of following links from initial adopters. This is based on the empirical observation that the likelihood of adoption of a user increases as the number of adopters in its neighbourhood increases ([Bakshy et al. 2009] ,[Romero et al. 2011]).

For evaluation the metric used is *Precision*@k, for $k = 10$, i.e., the fraction of candidates in the top-k list which have been correctly predicted as adopters. Average of the *Precision*@10 values for the topics in test set are reported.

4.2.3 Results

We compare the prediction performance of the proposed approach with the baselines in Table 4.2. It is observed that with increase in the number of initial adopters, the accuracy of prediction also increases in most of the cases. Moreover, the proposed method outperforms the baselines. Also, there is an increase in performance when vectors of dimension 300 are used, indicating that the added dimensions increases the amount of information captured by the user vectors. Figure 4.1 plots histogram of Precision@10 values for $n = 10$. There is a large variation in predictive performance among topics.

Model/ No. of initial adopters (Precision@10)	**n=10**	**n=20**	**n=30**	**n=40**	**n=50**
Frequency rank	0.089	0.089	0.088	0.087	0.086
Exposure rank	0.174	0.195	0.19	0.20	0.193
User vectors (100-dimensional vector)	0.303	0.304	0.317	0.312	0.32
User vectors (300-dimensional vector)	0.352	0.377	0.394	0.386	0.398

TABLE 4.2: Results of adopter prediction task, n is number of initial adopters

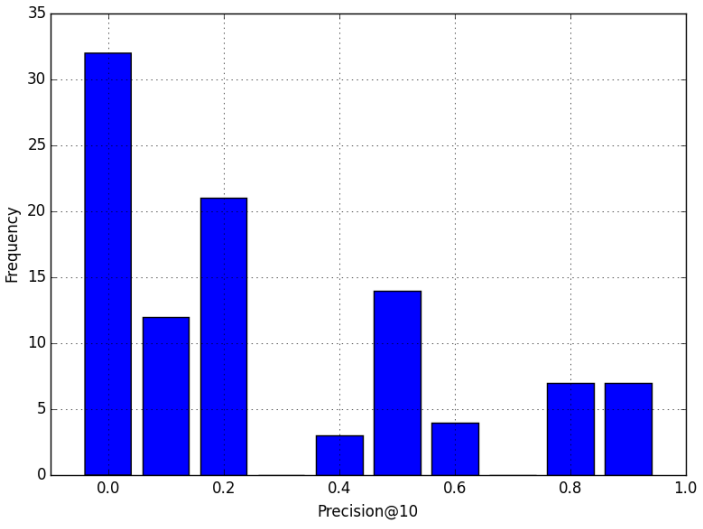


FIGURE 4.1: Histogram of Precision@10 values for 100 topics, $n = 10$

Chapter 5

Conclusion

The study of information diffusion in online social networks is of great interest. With the increased ability to observe user interactions in these online platforms, robust models of spread of information are possible.

In this project, we study two aspects of information diffusion. Firstly, we look at the phenomenon of virality, where certain topics manage to spread rapidly to a wide audience. We analyse different properties of the spread of a topic such as its network structure related properties, geographical distribution and textual content. The significance of these characterisations of the spread for predicting future popularity of the topic is studied. It is found that the properties quantifying the changes in the network structure of the spread such as conductance are very informative of topic's future popularity.
Secondly, we look at the problem of predicting future adopters of the topic in contrast to looking at the future size of the spread. We propose a method that automatically extracts features for users by using the timestamps of their adoptions. The extracted features for each user allows us to predict the future adopters of a topic based on its early spread. The main contribution of this work is that our methodology does not assume any knowledge of the diffusion network and instead leverages the information present in the topic adoption sequences. We also demonstrate adaptability of the learned features for the task of predicting geo-location of users. This suggests that the use of feature learning methods for extracting features for multiple prediction tasks in social network analysis is a promising direction of work.

5.1 Future Work

The future directions to explore include,

- The role of external influences (such as mass media, other social networks) on the spread of a topic and its virality can be investigated.
- The context for a user is defined based on the adoption timestamps only. Other criteria such as friendship information, location can be used to refine the context.
- We currently pool together tweets tagged by users with a hashtag, to consider them for obtaining context. However, topics can be inferred from content of the message to include more tweets that are related to a hashtag.
- The learned feature representations of users can be used with clustering methods to obtain topic-interest based communities.

Bibliography

[Weng et al. 2013] Weng, Lilian, Filippo Menczer, and Yong-Yeol Ahn. "Virality prediction and community structure in social networks." Scientific reports 3 (2013).

[Ardon et al. 2013] Sebastien Ardon, Amitabha Bagchi, Anirban Mahanti, Amit Ruhela, Aaditeshwar Seth, Rudra Mohan Tripathy, Sipat Triukose. Spatio-Temporal and Events-based Analysis of Topic Popularity in Twitter. (CIKM '13), pp 219-228, November 2013.

[Cheng et al. 2014] Cheng, Justin, Lada Adamic, P. Alex Dow, Jon Michael Kleinberg, and Jure Leskovec. "Can cascades be predicted?." In Proceedings of the 23rd international conference on World wide web, pp. 925-936. ACM, 2014.

[Guille et al. 2013] Guille, Adrien, Hakim Hacid, Cécile Favre, and Djamel A. Zighed. "Information diffusion in online social networks: A survey." ACM SIGMOD Record 42, no. 2 (2013): 17-28.

[Ma et al. 2013] Ma, Zongyang, Aixin Sun, and Gao Cong. "On predicting the popularity of newly emerging hashtags in twitter." Journal of the American Society for Information Science and Technology 64.7 (2013): 1399-1410.

[Tsur and Rappoport 2012] Tsur, Oren, and Ari Rappoport. "What's in a hashtag?: content based prediction of the spread of ideas in microblogging communities." In Proceedings of the fifth ACM international conference on Web search and data mining, pp. 643-652. ACM, 2012.

[Romero et al. 2011] Romero, Daniel M., Brendan Meeder, and Jon Kleinberg. "Differences in the mechanics of information diffusion across topics: idioms, political hashtags, and complex contagion on twitter." In Proceedings of the 20th international conference on World wide web, pp. 695-704. ACM, 2011.

[Weng et al. 2014] Lilian Weng, Filippo Menczer, and Yong-Yeol Ahn. "Predicting meme virality in social networks using network and community structure." In Proc. AAAI Intl. Conf. on Weblogs and Social Media (ICWSM), pp. 535-544. 2014.

[Romero et al. 2013] Romero, Daniel M., Chenhao Tan, and Johan Ugander. "On the interplay between social and topical structure." arXiv preprint arXiv:1112.1115 (2011).

[Bora 2014] Siddharth Bora. "Predicting Virality of Topics in Online Social Networks". Master's thesis, Indian Institute of Technology, Delhi, 2014

[Breiman 2001] Breiman, Leo. "Random forests." Machine learning 45, no. 1 (2001): 5-32.

[Hulse et al. 2007] Van Hulse, Jason, Taghi M. Khoshgoftaar, and Amri Napolitano. "Experimental perspectives on learning from imbalanced data." In Proceedings of the 24th international conference on Machine learning, pp. 935-942. ACM, 2007.

[Chawla et al. 2011] Chawla, Nitesh V., Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. "SMOTE: synthetic minority over-sampling technique." Journal of artificial intelligence research (2002): 321-357.

[Mehrotra et al. 2013] Mehrotra, Rishabh, Scott Sanner, Wray Buntine, and Lexing Xie. "Improving lda topic models for microblogs via tweet pooling and automatic labeling." In Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval, pp. 889-892. ACM, 2013.

[Mikolov et al. 2013a] Mikolov, Tomas, Kai Chen, Greg Corrado, and Jeffrey Dean. "Efficient estimation of word representations in vector space." arXiv preprint arXiv:1301.3781 (2013).

[Go et al. 2009] Go, Alec, Richa Bhayani, and Lei Huang. "Twitter sentiment classification using distant supervision." CS224N Project Report, Stanford (2009): 1-12.

[Perozzi et al. 2014] Perozzi, Bryan, Rami Al-Rfou, and Steven Skiena. "Deepwalk: Online learning of social representations." In Proceedings of the 20th

ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 701-710. ACM, 2014.

[Mikolov et al. 2013b] Mikolov, Tomas, Ilya Sutskever, Kai Chen, Greg S. Corrado, and Jeff Dean. "Distributed representations of words and phrases and their compositionality." In Advances in neural information processing systems, pp. 3111-3119. 2013.

[LeCun et al. 2015] LeCun, Yann, Yoshua Bengio, and Geoffrey Hinton. "Deep learning." Nature 521, no. 7553 (2015): 436-444.

[Bourigault et al. 2014] Bourigault, Simon, Cedric Lagnier, Sylvain Lamprier, Ludovic Denoyer, and Patrick Gallinari. "Learning social network embeddings for predicting information diffusion." In Proceedings of the 7th ACM international conference on Web search and data mining, pp. 393-402. ACM, 2014.

[J. Tang et al. 2014] Tang, Jian, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. "LINE: Large-scale Information Network Embedding." In Proceedings of the 24th International Conference on World Wide Web, pp. 1067-1077. International World Wide Web Conferences Steering Committee, 2015.

[L. Tang et al. 2009] Tang, Lei, and Huan Liu. "Relational learning via latent social dimensions." In Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 817-826. ACM, 2009.

[Kempe et al. 2003] Kempe, David, Jon Kleinberg, and Éva Tardos. "Maximizing the spread of influence through a social network." In Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 137-146. ACM, 2003.

[Fan et al. 2008] Fan, Rong-En, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. "LIBLINEAR: A library for large linear classification." The Journal of Machine Learning Research 9 (2008): 1871-1874.

[Davis et al. 2011] Davis Jr, Clodoveu A., Gisele L. Pappa, Diogo Rennó Rocha de Oliveira, and Filipe de L Arcanjo. "Inferring the location of twitter messages based on user relationships." Transactions in GIS 15, no. 6 (2011): 735-751.

[Bengio et al. 2013] Bengio, Yoshua, Aaron Courville, and Pierre Vincent. "Representation learning: A review and new perspectives." Pattern Analysis and Machine Intelligence, IEEE Transactions on 35, no. 8 (2013): 1798-1828.

[Bakshy et al. 2009] Bakshy, Eytan, Brian Karrer, and Lada A. Adamic. "Social influence and the diffusion of user-created content." In Proceedings of the 10th ACM conference on Electronic commerce, pp. 325-334. ACM, 2009.

[Myers et al. 2014] Myers, Seth A., and Jure Leskovec. "The bursty dynamics of the twitter information network." In Proceedings of the 23rd international conference on World wide web, pp. 913-924. ACM, 2014.

[Van der Maaten et al. 2008] Van der Maaten, Laurens, and Geoffrey Hinton. "Visualizing data using t-SNE." Journal of Machine Learning Research 9, no. 2579-2605 (2008): 85.

[Goyal et al. 2010] Goyal, Amit, Francesco Bonchi, and Laks VS Lakshmanan. "Learning influence probabilities in social networks." In Proceedings of the third ACM international conference on Web search and data mining, pp. 241-250. ACM, 2010.