# Opinion Mining on Twitter Data using Unsupervised Learning Technique

Muqtar Unnisa Deccan College of Engineering and Technology Darussalam Hyderabad TS Ayesha Ameen Associate Professor IT Dept Deccan College of Engineering and Technology Darussalam Hyderabad TS Syed Raziuddin, PhD Professor & HOD of CSE Dept Deccan College of Engineering and Technology Darussalam Hyderabad TS

# ABSTRACT

Social media is one of the biggest forums to express opinions. Sentiment analysis is the procedure by which information is extracted from the opinions, appraisal and emotions of people in regards to entities, events and their attributes. Sentiment analysis is also known as opinion mining. Opinion mining is to analyze and cluster the user generated data like reviews, blogs, comments, articles etc. These data find its way on social networking sites like twitter, facebook etc. Twitter has provided a very gigantic space for prediction of consumer brands, movie reviews, democratic electoral events, stock market, and popularity of celebrities.

The main objective of opinion mining is to cluster the tweets into positive and negative clusters. An earlier work is based on supervised machine learning (Naïve bayes, maximum entropy classification and support vector machines). The proposed work is able to collect information from social networking sites like Twitter and the same is used for sentiment analysis. The processed meaningful tweets are cluster into two different clusters positive and negative using unsupervised machine learning technique such as spectral clustering. Manual analysis of such large number of tweets is impossible. So the automated approach of unsupervised learning as spectral clustering is used. The results are also visualized using scatter plot graph and hierarchical graph.

## Keywords

opinion mining; feature extraction; feature vector; spectral clustering; k-means clustering; hierarchical clustering.

# 1. INTRODUCTION

The emergence of social media has given web users a venue for expressing and sharing their thoughts and opinion on different topics and events. Twitter, with nearly 600 million users and over 250 million messages per day, has quickly become a gold mine for organizations to monitor their reputation and brand by extracting and analyzing the sentiments of the tweets posted by public about them, their markets, and competitors [22]. Opinion mining or sentiment analysis [21] has recently received a lot of attention in the natural language processing (NLP) community. Opinion mining whose goal is to determine whether the opinion expressed in a twitter is "thumbs up" or "thumbs down" is arguably one of the most popular tasks in document level sentiment analysis. Opinion mining uses some algorithm techniques to cluster the user opinions into positive and negative clusters. Earlier work is based on supervised learning such as (naïve bayes, maximum entropy and SVM). Supervised learning have been popularly used and proven its effectiveness in sentiment classification. It is highly depend on large amount of labeled data which results in time

consuming and also expensive one. Based on the previous work unsupervised learning method are proposed to overcome the problem of supervised learning method which require large amount of unlabeled data. [11] Unsupervised learning is the machine learning task of inferring a function to describe the hidden structure from unlabeled data. Approaches to unsupervised learning are clustering (e.g., spectral clustering, k-means clustering and hierarchical clustering). Main steps included in this project are: extraction, pre-processing, unsupervised learning like spectral clustering [24] and finally k-means and hierarchical.

# 2. LITERATURE SURVEY

Opinion mining or sentiment analysis refers to the application of natural language processing, computational linguistic and text analytics to identify and extract subjective information in source materials. Millions of people have primary focus on social media platforms to share their own thoughts and opinions in regards to their day to day life, business, celebrity, entertainment, politics etc.

# 2.1 Twitter corpus

This thesis uses a dataset formed of collected messages from twitter. Twitter [2] contains a very large number of very short messages of 140 character created by the users of this micro blogging platform. The contents of the messages vary from personal thoughts to public statements [5]. Extracting the public opinion from social media text provides a challenging and rich context to explore computational models of natural language, motivating new research in computational linguistics.

## 2.2 Feature extraction

In opinion mining task documents and example are represented by thousands of tokens, which make the clustering problem very hard for many clustering system. In feature extraction [13], the original features converted to more compact new space. All the original features are transformed into new reduced space without deleting them but replacing the original features through a smaller representative set.

Feature selection [14] is a process of removing the irrelevant and redundant features from a dataset in order to improve the performance of unsupervised learning algorithm in terms of accuracy and time to build the model [7]. Y.Mejova et al [8] in his research work proposed that we can use presence of each character, frequency of occurrence of each character, word which is considered as negation etc. as feature for creating feature vector.

# 2.3 Machine learning

Machine learning consists of supervised learning and unsupervised learning. Data mining algorithm have two major

International Journal of Computer Applications (0975 – 8887) Volume 148 – No.12, August 2016

functions: classification and clustering. Classification maps data into predefined groups or classes and is often referred to as **supervised learning** [19] because the classes are determined before examining the data. Classification creates function from training data. The training data consists of pair of input object and desired output. Clustering is similar to classification except that the groups are not predefined, but rather defined by the data alone. Clustering is alternatively referred as **unsupervised learning** [15]. It can be thought of as partitioning or segmenting the data into groups that might or might not be disjoined. The clustering is usually accomplished by determining the similarity among the data on predefined attributes. The most similar data are grouped into clusters.

This paper presents an unsupervised learning algorithm for clustering tweets as recommended (thumps up) or not recommended (thumps down) by using a most popular clustering algorithm called spectral clustering. The most commonly used clustering algorithms are spectral clustering, hierarchical, and partitioning clustering algorithm [15]. Spectral clustering has been extensively used in many areas, including in the statistics, machine learning, pattern recognition, data mining, and image processing. There are many clustering algorithm that do a good job. However, recently spectral clustering techniques for data clustering have emerged as a powerful tool for clustering data [9]. Spectral clustering refers to a class of technique which relies on the Eigen structure of a similarity matrix. Clusters are formed by partition data points using similarity matrix [16].

K-means clustering is an idea, in which there is need to classify the given data set into K clusters; the value of K (number of cluster) is defined by the user which is fixed. K-means clustering algorithm is easy to understand and performance depends on initial centroid that's why algorithm doesn't guarantee for optimal solution [20].

Hierarchical clustering algorithm groups the data objects to form a tree shaped structure [16]. It can be broadly classified into agglomerative and divisive hierarchical clustering.

## 3. PROPOSED SYSTEM

Various techniques have been used to do sentiment analysis or opinion mining of tweets. The proposed system contains various phase of development. A dataset is created using twitter posts of movie reviews. As we know that tweets contains slang words and misspelling. So we perform a sentiment level sentiment analysis on tweets. This is done in three phases. In the first phase preprocessing is done. Then feature vector is created using relevant features. Finally using different unsupervised learning techniques, tweets are cluster into positive and negative classes. The supervised approach can be categorized as corpus-based methods as it uses labeled data to train sentiment classifiers. Given the difficulties of supervised sentiment analysis, it is conceivable that unsupervised approach [26] to sentiment classification is even more challenging. Unsupervised learning overcomes these difficulties. It divides the document into sentences and categorizes each sentence using word list of each category. Unsupervised learning does not require training set and test set.



Fig. 3: System model for opinion mining on twitter data.

In the proposed system architecture (Fig. 3), it is shown that the social networking site twitter has been the source for data collection. Database is created using twitter posts of movie reviews and related tweets about those movies. The preprocessing is the major part of this architecture. The collected reviews are preprocessed and then words are extracted. Feature extraction are taken for absolute positive words like "wonderful", "awesome", "always" etc and negative words such as "never", "not", "hardly" etc. Feature vector is the vector of all high information words which appear in the document. Spectral clustering is the clustering techniques which cluster the tweets into positive and negative clusters. Algorithm- I explains the data flow of the system.

Algorithm – I :Opinion Mining On Twitter With Unsupervised Learning

Input: Enter the set of tweets related to movie reviews

Output: Outcome in terms of clusters

- 1. T = { tweeted, tweet Text, userId, tweet Date} Extracted from twitter and store it in excel or notepad.
- 2. T = normalized (T). Perform data cleaning and normalization.
- 3. Extract the extended targets from T. Calculate frequency of words TF-IDF and store extended targets and features. Create dictionary H high information words.
- 4. FV = word features (T) create feature vector of word set.
- 5. Use test dataset to cluster using unsupervised learning methods.
- 6. FV' = SC (FV). Perform spectral clustering of feature vector.
- Provide the incremental training with subset of FV' as T<sub>i</sub> with output class C; C ∈ {positive, negative} and i is iteration.
- 8. After the spectral clustering now apply k-means to normalization matrix N.
- Finally apply hierarchical clustering algorithm to merge the cluster into one cluster

Following subsections explain the each Figure 3 blocks in detail

## 3.1 Data extraction

Data extraction is to fetch the tweets using Twitter API v 1.1.to collect the data from various hash-tags like #moviereviews. We collected approximately 2000 tweets for movie reviews. In order to have access to twitter data programmatically, this need to create an app that interacts with the twitter API. Sign in to twitter it will provide us with consumer key and consumer secret key: these are the application settings that should be kept private. From the configuration page of app, it can also require an access token and an access token secret. Similarly to the consumer keys, these strings must also be kept private: they provide the application access twitter on behalf of an account.

## 3.2 Data cleaning and normalization

In order to remove stop words and extract features we perform data cleaning and normalization [25]. Preprocessing is the major part of this thesis. Preprocessing of data is the process of preparing and cleaning the tweets for clustering. Reducing the noise in the text should help to improve the performance of the clustering and speed up the clustering process. This project performed the following operations on tweets during cleaning and normalization.

- Tokenization: given input as character sequence, tokenization is a task of chopping it up into pieces called tokens and at the same time removing certain characters such as punctuation marks.
- Stop words removal: A stop-list is the name commonly given to a set or list of stop words. It is typically language specific, although it may contain words. Some of the commonly used stop words from English include "a", "of", "the", "I", "it", "you", "and", these are generally regarded as functional words which do not carry any meaning.
- Stemming: it is a process for reducing derived words to their stem, or root form. Stemming program are commonly referred to as stemmers or stemming algorithms. e.g. "developed", "development", "developing" is reduced to the stem "develop".
- Internet Acronyms And Emoticons: in this process it convert internet acronyms like <3 to "love" or "gud" to "good" in order to make the meaning out of the symbols posted as part of the tweets.
- Word Expansion: this system expands the famous acronyms as well. The expansion is considered for Standard English words. As an example AFAIK-As Far As I Know, LOL-lots of laugh, etc.
- Repeated Words: if a word is being repeated in a tweet for more than two times consecutively, occurrences of the word had been limited to two occurrences. E.g. very very very very very good has been replaced by very very good. And a character like "loonnnnggggg" has been replaced by "long".

#### **3.3 Feature extraction**

Features from tweets are extracted and this project uses the unigram, bigram and unigram + bigram (hybrid) feature extraction method. Hybrid features are taken for absolute positive words like "wonderful", "awesome", "always" etc and negative words such as "never", "not", "hardly" etc. As an example in following tweet user explains positive sentiment for target "movie".

#### "fan movie was fantastic sharukh acting was fabulous."

This also ignores the future transitive verbs when followed by the query terms e.g. "sharukh" does not make any value for prediction. This project also makes sure that generic terms like "sharukh", "movie" are ignored during feature extraction to ensure exclusion of non-subjective data. The examples of feature words extracted from sample tweets are shown below.

#### Table 1. Example showing tweets and feature words

Positive Tweets	Feature Words
Bajrangi bhaijann The film is exceptionally positive. Celebrate humanity. Doesn't take any religion or country's side.	'positive', 'Humanity', 'religion', 'country's', 'slide'

Negative Tweets	Feature Words
AT_USER disappointed. Watched a movie. It is a waste of time.	'disappointed', 'Watched', 'movie', 'waste'
I miss my mom and dad. I hate this life	miss', 'hate'

## **3.4 Feature selection**

Feature selection [25] is used to make the clustering more efficient by reducing the amount of a data to be analyzed as well as identifying relevant features to be considered in clustering process.

#### **3.5 Feature vector**

The first step in modeling the document into vector space is to create a dictionary of terms present in the documents. To do this, we need to select all terms from the document and convert it to a dimension in the vector space. Tf-idf term frequency and inverse document term frequency value increases proportionally to the number of times a word appears in the document.

Let's take the documents below to define our document space:

Train Document Sets:

D1: The sky is blue.

D2: The sun is bright.

Test Document Set:

D3: The sun in the sky is bright.

D4: We can see the shining sun, the bright sun.

Now what we have to do is to create an index vocabulary (dictionary) of the words of the train document set, using the document d1 and d2 from the document set, we 'ill have the following dictionaries of all the words E (t) where t is the term:

$$\mathbf{E}(\mathbf{t}) = \begin{cases} \text{If } t \text{ is "blue"} \\ \text{If } t \text{ is "sun"} \\ \text{If } t \text{ is "bright"} \\ \text{If } t \text{ is "sky"} \end{cases}$$

Note that the terms like "is" and "the" were ignored as cited before. Now we can convert the test document set into a vector space, the first term of the vector represent "blue" term of our vocabulary, the second represents "sun" and so on. Now use the term-frequency to represent each term in our vector space; the term-frequency is nothing more than a measure of how many times the term present in our vocabulary E (t) are present in the document d3 or d4, this define the term-frequency as a counting function:

Tf (t, d) =  $\sum_{x \in d} (x, t)$ 

This function counts the frequency of a word i.e. *tf* (*t*, *d*) returns is how many times that word present in the document, In this thesis we use a threshold value  $\theta=2$  if the word is appearing in the document more than two times that can be taken as a high information word and is taken into a dictionary making the dictionary of all high information words.

Where the fr(x, t) is a simple function defined as:

$$Fr(x, t) = \begin{cases} 1, & \text{if } x=t \\ 0, & \text{otherwise} \end{cases}$$

Now it compares the documents term with high information word dictionary if the term occur in the document it returns the value 1 in the vector space else its return 0.its make the feature vector of 0's and 1's.

## **3.6 Spectral clustering**

Spectral clustering is a more powerful and specialized clustering algorithm, spectral clustering technique reduces dimensions using Eigen value of the similarity matrix of the data. The similarity matrix is provided as input and consists of a quantitative evaluation of the relative similarity of each pair of points in the dataset.

The spectral clustering algorithm is an algorithm for grouping N data points in an I-dimensional space into several clusters. Each cluster is parameterized by its similarity, which means that the points in the same group are similar and points in the different group are dissimilar to each other.

Spectral clustering is more advanced algorithm compared to k-means as its uses several mathematical concepts (i.e. similarity matrices, similarity graph, graph Laplacian, Eigen values and Eigen vectors) in order to divide similar data points in the same group and dissimilar data points in different group. This spectral clustering [31] works well for many real world data sets, this need some modification in terms of improving its time complexity and space complexity.

Spectral clustering algorithm can be summarizing as follow in Algorithm -2 [31]

#### Algorithm-2: Spectral clustering

**Input**: given a set of points  $S = \{S_1, \dots, S_n\}$  that this Project wants to cluster into k subset.

Output: positive and negative clusters.

Step1: Form the Affinity matrix (similarity matrix)  $A \in \mathbb{R}^{n \times m}$  defined by

 $A_{ij} = \exp(-||s_i - s_j||^2 / 2\sigma^2)$  if  $i \neq j$ , and  $A_{ii} = 0$ .

Step 2: Define D to be the diagonal matrix whose (i, i) element is the sum of A's i-th row and construct the matrix

 $L = D^{-1/2}AD^{-1/2}.$ 

Step 3: Find the eigenvalues and eigenvectors of L.

 $L\vartheta_i = \lambda \vartheta_i$ 

Where L is a square matrix  $\vartheta$  is eigenvectors  $\lambda$  is eigenvector.

Step 4: Create a new matrix from the m eigenvectors that correspond to the m largest eigenvalues.  $X = \{x_1, x_2...x_k\} \in \mathbb{R}^{n \times m}$  by stacking Eigen vectors in column.

Step 5: Each data point is now rank-reduced to a point in the m-dimensional space. Normalize each point to unit length (while retaining the sign of each value). Form matrix Y from X.

$$Y_{ij} = X_{ij} / (\sum_j X_{ij}^2)^{1/2}).$$

Step 6: Cluster the resulting data points using k-means.

In these steps we use spectral clustering to identify unambiguous reviews. To make use of spectral clustering, we first create similarity matrix, defining the similarity between two tweets as the dot product of the feature vectors, but following to Ng et al (2002), set its diagonal entries to 0. Then perform an Eigen decomposition of this matrix. Finally using resulting eigenvectors, partition the normalized reviews into two sets. To create two clusters the most common way is to use only the second highest eigenvector induces an intuitively ideal partition of the data the partition induced by the minimum normalized cut of the similarity graph, where the nodes are the data points and the edge weights are the pair wise similarity values of the points. Clustering in one dimensional space is trivial: all is need to determine a threshold for partitioning the points. A common approach is to set the threshold to zero. In other words, all points whose value in the second Eigen vector is positive are classified as positive, and the remaining points are classified as negative. To see how this is done, consider the example in figure 4.1, where the goal is to produce two clusters from five data points.

/11100\	/-0.6983 0.7518 ∖
11100	-0.6983 0.7518
00110	-0.9869 - 0.1616
00011	-0.6224 - 0.7827
00011/	-0.6224 - 0.7827/

# Fig 2: sample data and the top two eigenvectors of its Laplacian

In the matrix on the left, each row is the feature vector generated for  $D_i$ , the i-th data point. By inspection, one can

identify two clusters,  $\{D_1, D_2\}$  and  $\{D_4, D_5\}$ .  $D_3$  is ambiguous, as it bears resemblance to the points in both clusters and therefore can be assigned to any of them. In the matrix on the right, the two columns correspond to the top two eigenvectors obtained via an Eigen decomposition of the Laplacian matrix formed from the five data points. As we can see the second eigenvector gives us a natural cluster assignment: all the points whose corresponding values in the second eigenvector are strongly positive will be in one cluster, and the strongly negative points will be in another cluster. Being ambiguous  $D_3$  is weekly negative and will be assigned to the "negative" cluster.

This algorithm can be thought of as opposite to self training. In self training, its iteratively train the classifier on the data labeled so far, use it to classify the unlabeled instances, and augment the labeled data with most confidently labeled instance.

## **3.7 K-means clustering**

The k means algorithm is used as a last step in spectral clustering algorithm to plot the graph. K-means is the best known partitioning clustering algorithm due to its simplicity and efficiency. Given the data points and required number of k cluster (k is specified by the user), this algorithm iteratively partitions the data into k clusters based on distance function.

K-means algorithm summarizes as follow in the following Algorithm-3

Algorithm – 3: k-means algorithm

**Input**: Let  $X = \{x_1, x_2, x_3, \dots, x_n\}$  bet the set of data points and  $V = \{v_1, v_2, v_3, \dots, v_c\}$  bet the set of centers. **Output**: clusters

Step 1: Randomly select 'c' cluster centers.

Step 2: Calculate the distance between each data point and cluster centers.

Step 3: Assign the data point to the cluster center whose distance from the cluster center is minimum of the entire cluster centers.

Step 4; Recalculate the new cluster center using:

 $V_i = (1/C_i) \sum_{i=1}^{c_i} x_i$ 

Where 'C<sub>i</sub>' represents the number of data points in i-th cluster.

Step 5: Recalculate the distance between each data points and new obtained cluster centers.

Step 6; if no data point was reassigned then stop, otherwise repeat step 3.

## **3.8 Hierarchical clustering**

Hierarchical clustering (HC) is also another way to plot the graph based on data points. HC is a method of cluster analysis which seeks to build a hierarchy of clusters. HC is generally falling into two types: Agglomerative (bottom up) and Divisive (top down).

In Agglomerative approach each observation start in its own cluster, and pairs of clusters are merged as one move up the hierarchy.

In Divisive approach all observation start in one cluster and splits are performed recursively as one move down the hierarchy.

Hierarchical clustering algorithm is summarizes as follow in Algorithm-4

Algorithm-4: *Hierarchical clustering agglomerative algorithm* 

**Input**: Let  $X = \{x_1, x_2, x_3, \dots, x_n\}$  bet the set of data points

**Output**: hierarchical graph

Step 1: begin with the disjoint clustering having level L(0) = 0 and sequence number m=0.

Step 2: find the least distance pair of clusters in the current clustering, say pair (r), (s), according to d[(r)(s)] = mind[(i)(j)] where the minimum is over all pairs of clusters in the current clustering.

Step 3: increment the sequence number: m=m+1, merge clusters (r) and (s) into single cluster to form the next clustering m. set the level of this clustering to L(m) = d[(r),(s)].

Step 4: update the distance matrix D, by deleting the rows and columns corresponding to clusters (r) and (s) and adding a row and column corresponding to the newly formed cluster. The distance between the new cluster and the old cluster (k) is defined in this way:

d[(k)(r, s)] = min(d[(k), (r, s)], d[(k), (s)]).

Step 5: if all the data points are in one cluster then stop, else repeat from step 2.

## 4. EXPERIMENTAL RESULT

This thesis provides experimental results to validate the usefulness of the results presented in previous sections. In this research first illustrate "spectral clustering" of feature vector with k-means and hierarchical clustering.

Using Eigen vectors to initialize k-means give better initial and final objective function values and better clustering results. Thus the theoretical connection between spectral clustering and k-means helps in obtaining higher quality results.

Finally this shows that distance estimation technique save a considerable amount of computation time, verifying the scalability of our approach

This screenshot (Fig 3) shows the data extracted (Tweets files). These are the screenshots, which are taken during the project execution, and each screenshot shows project module working.



#### Fig 3 Data extracted from twitter

This screenshot (Fig 4) is the data cleaning and normalization, which shows the data after preprocessing and word frequencies.

"wooden": 1.9377894027268239,
"raoul": 2.0647876744394846,
"flashy": 0.0010077812490600657,
"stereotypical": 2.0647876744394846
"precursor": 2.0647876744394846,
"scheider": 6.196103501830205,
"jeff": 1.9377894027268239,
"force": 0.7321444935586547,
"pragmatic": 2.0647876744394846,
"warns": 2.0647876744394846,
"bubbling": 1.9377894027268239,
"direct": 4.130155426888364,
"pulse": 1.9377894027268239,
"depraved": 1.9377894027268239,
"likely": 0.6051105381506543,
"street": 0.0020161289060161886,
"loathing": 1,9377894027268239.

#### Fig 4: Word frequencies

	-						-										•						-					
SIM	IL	AF	LI:	ΓY		MΆ	TF	LI)	ĸ																			
[0,	0		8	85	0	72	72	70	61	62	28	74	Ε,		٥.	1	06	54	89	93	32	21	0	23	5	54	9,	
52,	0		4:	20	3	23	77	40	00	70	00	65	5,		٥.	0	68	31	99	94	33	39	4	70	4	73	5,	
435		ο.	10	02	0	88	44	18	83	81	11	25	έ,		٥.	1	00	)3	8	58	48	86	3	98	7	39	з,	
[0.	08	85	0	72	7	27	61	62	28	74	ι,	C	),		٥.	0	71	2	4(	03	53	35	2	18	9	66	6,	
, 0	.0	49	99	89	7	99	04	10	00	37	11	,	0		08	1	11	0	71	10	5(	65	3	81	2	7,	0	э.
в,	ο.	06	6	22	6	61	78	153	32	52	21	9,		0	. 1	1	02	20	71	75	31	75	5	59	6	76		C
[0.	10	64	8	93	3	21	02	:35	55	49	۰,	C	٥.	0	71	2	40	)3	53	35	21	18	9	66	6	,	ο,	
21,	0		6	01	4	62	47	23	59	75	54	49	۶,		٥.	0	97	15	9(	00	01	72	9	48	5	33	з,	
963		ο.	1:	32	8	03	17	88	31	49	93:	26	54	,	0		08	88	39	99	13	39	2	17	5	72	78	з,
[0.	08	30	0	97	0	75	15	29	92	05	έ,	C	).	0	98	7	24	18	1:	29	59	98	4	87	4	,	ο.	. c
248		ο.	0	85	7	32	05	72	27	47	13	99	94	,	0		06	50	8	58	0(	61	9	45	0	18	46	ā,
894		ο.	0	91	0	99	06	55	74	99	99:	14	4	,	0		06	58	9(	08	18	85	8	82	8	93	57	7,
٢٥.	08	38	9	05	4	34	48	98	32	32	2,	0	).	0	58	6	89	96	50	03	82	23	3	36	1		ο.	. 1

Fig 5: Similarity matrix

lai	rgest	2	eigen	valu	ies a	and	vectors
2.7	78679	757	115				
0.3	30897	248	0035				
11	0.13	945	264	0.188	3713	34]]	
11	0.13	954	669	0.227	7055	08]]	
11	0.17	452	286	0.167	7067	29]]	
11	0.15	035	965	0.324	1552	81]]	
11	0.14	599	869	0.241	131	66]]	
11	0.13	460	098	0.451	973	5 ]]	
11	0.19	161	99	0.222	2580	64]]	
11	0.13	550	966	0.250	0533	51]]	
11	0.09	519	594	0.408	8916	61]]	
11	0.12	359	315	0.143	3425	07]]	

Fig 6: Two largest eigen values and eigen vectors

NOF	RMALIZED MAT	TRIX
] ]	2.21535143	0.92068509]]
1 J	2.2168455	1.10774481]]
] ]	2.77247867	0.81507943]]
1 J	2.38862065	1.58341181]]
] ]	2.31934235	1.1764209 ]]
] ]	2.13827779	2.20506541]]
] ]	3.04408303	1.08591512]]
] ]	2.15271304	1.2222902 ]]
1 J	1.51228732	1.99500164]]
] ]	1.96340682	0.69973496]]
1 J	2.62558335	0.43878512]]
	4 55300000	0.0770000011

#### Fig 7: Normalized matrix

Finally, after normalization matrix next step is to plot the graph with positive and negative clusters red indicate the negative clusters and blue as positive clusters with k-means as shown in Fig-8.



Fig 8: Cluster review

This is an another way to plot the graph with hierarchical clustering shows the hierarchy of cluster as shown in Fig 9.



Fig 9: Hierarchical clustering



Fig 10: Hierarchical clustering dendogram

## 5. RESULTS AND DISCUSSION

This thesis uses the dataset collected by twitter API containing tweets in English language, for the original terms and extended features This research achieves the best sentiment accuracy with spectral clustering which also leads us to the best accuracy result. Among feature extraction model, hybrid model of unigram + bigram provides best result. Figure-11 provides the result obtained with proposed method.

Spectral clustering outperforms SVM, Max Ent and Naïve Bayes for the process of feature presence verification for sentiment analysis and opinion mining. An important advantage of spectral clustering over supervised learning SVM obtains the clusters in a convex optimization problem by always finding global minimum with a unique solution; whereas supervised learning with SVM may not converge to an optimal/global solution.



Fig -11: sentiment accuracy comparison for SVM, SVM PCA and SC

In this thesis we accumulate 2000 tweets about movie reviews and found the large variation in result by the moderate level of accuracy for opinion mining or sentiment analysis of twitter data. Overall more number of positive tweets we find than negative value lesser accuracy towards the result as it can be seen from Fig-12.



Fig-12: Positivity and Negativity of opinions

## 6. CONCLUSION

Overall, we conclude that social network based behavioral analysis parameters can increase the prediction accuracy along with sentiment analysis. Twitter base social network provides the great platform in measuring the public opinion with the reasonable accuracy in movie reviews with spectral clustering based machine learning algorithms for sentiment analysis.

In this thesis, a new opinion mining of twitter data using unsupervised learning technique is proposed that can solve the problem of domain dependency and reduce the need of annotated training data. Unsupervised machine learning techniques have shown better performance than supervised learning. This project main goal is to overcome the problem of clustering multiple files with unlabeled data and perform sentiment classification. Unsupervised approaches seen promising in this regard since they do not require annotated training data, just access to sufficient raw text in each domain. Experimental results on two thousand tweets demonstrate that our approach is not specific to movie reviews and can be easily applicable to other domain with sufficiently large corpuses.

## 7. REFERENCES

- Influence factor based opinion mining of Twitter data using supervised learning Malhar Anjaria; Ram Mohana Reddy Guddeti 2014 Sixth International Conference on Communication Systems and Networks (COMSNETS) Year: 2014
- [2] Dewan Md. Farid, and Chowdhury Mofizur Rahman, "Mining Complex Data Streams: Discretization, Attribute Selection and classification," Journal of Advances in Information Technology, Vol. 4, No. 3, August 2013, pp. 129-135.
- [3] U. von Luxburg, A tutorial on spectral clustering [J]. Statistics and Computing, 2007, 17(4): 395–416.
- [4] Liu, B. (2010), "Sentiment Analysis and Subjectivity". Appeared in Handbook of Natural Language Processing, Indurkhya, N. & Damerau, F.J. [Eds.].
- [5] Twitter as a Corpus for Sentiment Analysis and Opinion Mining, Alexander Pak, Patrick Paroubek. (2014)
- [6] Parikh and Movassate, Sentiment Analysis of User-Generated Twitter Updates using Various Classification Techniques, Stanford University, 2009
- [7] Saeys, Y, Inza, I & Larrañaga, P 2007, "A review of feature selection techniques in bioinformatics. Bioinformatics", vol. 23, no. 19, pp.2507-2517
- [8] Y.Mejova, 'sentiment analysis: An overview', Y.Mejova/publications/CompsYelenaMejova, vol. 2010-02-03, 2009, 2009.
- [9] Sajib Dasgupta and Vincent Ng "Mine the Easy, Classify the Hard: A Semi-Supervised Approach to Automatic Sentiment Classification", *Human Language Technology Research Institute, University of Texas at Dallas.*
- [10] M Ashraf et. al. "Multinomial Naive Bayes for Text Categorization Revisited", University of Waikato
- [11] D. O. Computer, C. Wei Hsu, C. chung Chang, and C. jen Lin. A practical guide to support vector classification chih-wei hsu, chih-chung chang, and chih-jen lin. Technical report, 2003.
- [12] K. Nigam, J. Laverty, and A. Mccallum. Using maximum entropy for text classification. In IJCAI-99 Workshop on Machine Learning for Information Filtering, pages 61 to 67
- [13] J.C.Gomez, E. Boiy, M.F.Moens. Highly discriminative statistical features for email classification. Knowledge and Information System, (2012), 31(1); 23-53
- [14] A survey of machine learning techniques for sentiment classification mohini chaudhari and sharvari govilkar department of computer engineering, university of mumbai, piit, new panvel, india International Journal on Computational Science & Applications (IJCSA) Vol.5, No.3, June 2015
- [15] International Journal of Science, Engineering and Technology Research (IJSETR) Volume 2, Issue 4, April 2013, Survey Thesis on Clustering Techniques Amandeep Kaur Mann (M.TECH C.S.E) Department of Computer Science & Engineering of RIMT Institutions, Mandi Gobindgarh, Punjab, India.

- [16] A Survey On Partition Clustering Algorithms S. Anitha Elavarasi Lecturer, Department Of Cse, Sona College Of Technology, Salem-636 005, India, Vol. 1 Issue 1 January 2011
- [17] Agglomerative Hierarchical Clustering Algorithm- A Review K.Sasirekha, P.Baby Department of CS, Dr.SNS.Rajalakshmi College of Arts & Science, International Journal of Scientific and Research Publications, Volume 3, Issue 3, March 2013 1 ISSN 2250-3153
- [18] Int. J. Advanced Networking and Applications Volume: 03, Issue: 01, Pages: 1006-1011 (2011) Performance Analysis of Hierarchical Clustering Algorithm K.Ranjini Department of Computer Science and Engineering, Einstein College of Engineering, Tirunelveli, Ind
- [19] A Survey on Supervised Learning for Word Sense Disambiguation Abhishek Fulmari1, Manoj B. Chandak2 International Journal of Advanced Research in Computer and Communication Engineering Vol. 2, Issue 12, December 2013.
- [20] Ahamed Shafeeq BM and Hareesha K S, "Dynamic Clustering of Data with Modified K-Means Algorithm," proceeding of the 2012, International Conference on Information and Computer Networks (ICICN 2012).
- [21] Sentiment Analysis and Opinion Mining: A Survey, Volume 2, Issue 6, June 2012 ISSN: 2277 128X International Journal of Advanced Research in Computer Science and Software Engineering.
- [22] A Survey Paper on Twitter Opinion Mining Geetanjali S. Potdar1, Prof R. N. Phursule2 International Journal of Science and Research (IJSR) ISSN (Online): 2319-7064 Index Copernicus Value (2013): 6.14.
- [23] Eigenvalues and Eigenvectors: Formal, Symbolic and Embodied Thinking Michael O. J. Thomas The University of Auckland.
- [24] Spectral Clustering: Advanced Clustering Techniques 1 S. V. Suryanarayana (Ph.D), 2Guttula Rama Krishna (M.Tech), 3Dr. G. Venkateswara Rao (Ph.D) International Journal of Advanced Research in Computer Science and Software Engg. 4(11), November - 2014, pp. 625-62.
- [25] Feature selection and classification approach for sentiment analysis gautami tripathil and naganna s.2 machine learning and applications: an international journal (mlaij) vol.2, no.2, june 2015
- [26] www.ee.columbia.edu/unsupervisedlearning.pdf
- [27] (IJARAI) International Journal of Advanced Research in Artificial Intelligence, Vol. 2, No. 2, 2013 Comparison of Supervised and Unsupervised Learning Algorithms for Pattern Classification R. Sathya Professor.
- [28] https://en.wikipedia.org/wiki/Hierarchical\_clustering
- [29] https://en.wikipedia.org/wiki/K-means\_clustering
- [30] https://charlesmartin14.wordpress.com/2012/spectralclustering
- [31] A. Ng, M. Jordan, Y. Weiss, On spectral clustering: analysis and an algorithm, Adv. Neural Inf. Process. Syst. 14 (2001) 849–856.