A Methodological Framework for Statistical Analysis of Text from Social Media

Sophia Kleisarchaki

Thesis submitted in partial fulfillment of the requirements for the

Masters' of Science degree in Computer Science

University of Crete
School of Sciences and Engineering
Computer Science Department
Knossou Av., P.O. Box 2208, Heraklion, GR-71409, Greece

Thesis Advisor: Prof. *Vassilis Christophides*

This work has been performed at the **University of Crete, School of Science and Engineering, Computer Science Department and at the Institute of Computer Science of Foundation for Research and Technology - Hellas (FORTH)**

The work is supported by the **Institute of Computer Science (ICS) - Foundation of Research and Technology (FORTH)**

University of Crete
Computer Science Department

A Methodological Framework for Statistical Analysis of Text from Social Media

Thesis submitted by
Sophia Kleisarchaki
in partial fulfillment of the requirements for the
Masters' of Science degree in Computer Science

THESIS APPROVAL

Author: _____
Sophia Kleisarchaki

Committee approvals: _____
Vassilis Christophides
Professor, Thesis Supervisor

_____
Dimitris Kotzinos
Assistant Professor in the Department of Geoinformatics and Surveying, TEI of Serres, Committee Member

_____
Ioannis Tsamardinos
Assistant Professor, Committee Member

Departmental approval: _____
Angelos Bilas
Professor, Director of Graduate Studies

Heraklion, October 2012

Abstract

We are witnessing an unprecedented growth of interest in social media enabling people to achieve a *near real-time information awareness*. Several online networking sites (e.g. Facebook), micro-bloging applications (e.g. Twitter) and Social news (e.g. Digg) produce on a daily basis vast amounts of User-Generated Content (UGC) under the form of textual posts related to a wide variety of real-world news (personal, political, commercial, etc.). The automated analysis of such social text streams has already created scientific and business value. Several machine learning clustering methods have been proposed in this respect during the last years. However, there is still no commonly used methodology for statistical analysis of textual content produced in social media that take into account the peculiarities of social text stream. For instance, Twitter is overwhelmed by *low quality* posts (ungrammatical, spam etc), having a great impact on the size of the extracted vocabulary. Furthermore, the users' posts are *heterogeneous* and *noisy* ranging from personal stories to breaking news affecting the *number* and *utility* of recognized clusters. Also, posts are characterized by a non-stationary data distribution of a *highly evolving* behaviour that may causes evolution of the *shape*, *centroid* and *density* of the clusters. In this thesis, motivated by the aforementioned observations and the poor quality results of some well-known clustering algorithms we are interested in understanding which of the peculiarities of social text streams exhibited in reality affect the evolving behaviour of clusters automatically detected by various kinds of machine learning algorithms. In particular, within the scope of our dataset we have evidence that the clusters' centroid of a topic move in a multi-dimensional space indicating a shift of topic's interest discussed over time. Furthermore, the clusters' shape also changes over time indicating the users' opinion convergence or discrepancy. Based on such methodological framework, we plan to illustrate the weakness of several clustering algorithms proposed in the literature in order to adjust the analysed clusters to the peculiarities of social text streams and finally improve their clustering quality.

Περίληψη

Γινόμαστε μάρτυρες μιας απρόσμενης αύξησης του ενδιαφέροντος στα κοινωνικά μέσα που επιτρέπει στους χρήστες να επιτύχουν μια *σχεδόν πραγματικού χρόνου ενημέρωση*. Αρκετές σελίδες κοινωνικής δικτύωσης (π.χ. Facebook), ιστολόγια (π.χ. Twitter) και κοινωνικά μέσα ενημέρωσης (π.χ. Digg) παράγουν σε καθημερινή βάση μεγάλο όγκο από περιεχόμενο προερχόμενο από τον χρήστη υπό την μορφή κειμένου μηνυμάτων, σχετιζόμενα με ένα ευρύ φάσμα ειδήσεων του πραγματικού κόσμου (προσωπικές, πολιτικές, εμπορικές κτλ). Η αυτοματοποιημένη ανάλυση τέτοιου είδους κοινωνικών ροών κειμένου έχει ήδη δημιουργήσει επιστημονική και εμπορική αξία. Αρκετές μέθοδοι συσταδοποίησης μηχανικής μάθησης έχουν προταθεί στο πλαίσιο αυτό τα τελευταία χρόνια. Ωστόσο, δεν υπάρχει ακόμη μια κοινώς χρησιμοποιούμενη μεθοδολογία για τη στατιστική ανάλυση του κειμενικού περιεχομένου που παράγεται στα κοινωνικά μέσα ενημέρωσης η οποία να λαμβάνει υπόψη τις ιδιομορφίες των κοινωνικών ροών κειμένου. Για παράδειγμα, το Twitter κατακλύζεται από μηνύματα *χαμηλής ποιότητας* (σόλοικη σύνταξη, ανεπιθύμητα κτλ), προκαλώντας σημαντικό αντίκτυπο στο εξαχθέν λεξιλόγιο και στην αναπαράσταση της βάρυνσης του. Επιπλέον, τα μηνύματα των χρηστών είναι *ετερογενή* και *θορυβώδη* κυμαινόμενα από προσωπικές ιστορίες μέχρι έκτακτες ειδήσεις, επηρεάζοντας το *πλήθος* και την *ωφελιμότητα* των συστάδων. Τα μηνύματα χαρακτηρίζονται από μια μη στατική κατανομή δεδομένων *εξαιρετικά εξελισσόμενης* συμπεριφοράς που πιθανόν προκαλεί εξέλιξη στο *σχήμα*, το *κεντροειδές* και την *πυκνότητα* των συστάδων. Στην παρούσα διατριβή παρακινούμενοι από τις προαναφερθείσες παρατηρήσεις και τα πενιχρά αποτελέσματα μερικών γνωστών αλγορίθμων συσταδοποίησης ενδιαφερόμαστε να κατανοήσουμε ποίες από τις ιδιομορφίες των κοινωνικών ροών κειμένου που υπάρχουν στην πραγματικότητα επηρεάζουν την εξέλιξη της συμπεριφοράς των συστάδων οι οποίες εντοπίζονται αυτόματα από διάφορα είδη αλγορίθμων μηχανικής μάθησης. Ειδικότερα, εντός του πεδίου των δεδομένων μας έχουμε ενδείξεις ότι το κεντροειδές της συστάδας ενός θέματος κινείται μέσα σε ένα πολυδιάστατο χώρο υποδεικνύοντας μια μετατόπιση του θεματικού ενδιαφέροντος που συζητιέται στην πάροδο του χρόνου. Επιπλέον, το σχήμα των συστάδων επίσης αλλάζει με την πάροδο του χρόνου υποδεικνύοντας τη σύγκλιση ή απόκλιση των απόψεων των χρηστών. Βασιζόμενοι σε αυτό το μεθοδολογικό πλαίσιο, σκοπεύουμε να σκιαγραφήσουμε τις αδυναμίες αρκετών αλγορίθμων συσταδοποίησης, που προτάθηκαν στη βιβλιογραφία, να προσαρμόσουν τις αναλυόμενες συστάδες στις ιδιομορφίες των κοινωνικών ροών κειμένου και εν τέλει να βελτιώσουν την ποιότητα συσταδοποίησης τους.

Ευχαριστίες

Η παρούσα μεταπτυχιακή εργασία δε θα μπορούσε να ολοκληρωθεί χωρίς τη συμβολή του επόπτη καθηγητή μου κ. Χριστοφίδη Β. και των συνεπιβλέποντων καθηγητών κκ. Κοτζίνο Δ. και Τσαμαρδίνο Ι.

Αρχικά, ευχαριστώ θερμά τον κ. Χριστοφίδη για την επιστημονική τοποθέτηση του απέναντι στην παρούσα μελέτη, την επιμονή του στην ποιοτική δουλειά, την υποστήριξη του και το πολύ καλό κλίμα συνεργασίας που ανέπτυξε. Πέρα από εξαίρετος επιστήμονας, με παιδεία και νόηση που διοχετεύεται σε έναν αξιοθαύμαστο λόγο, είναι ένας υπέροχος άνθρωπος με μεγάλη κατανόηση και γενναιόδωρο χαρακτήρα.

Τον κ. Κοτζίνο για την καθοδήγηση του, την ψύχραιμη και διαυγή αντιμετώπιση όλων των επιμέρους ζητημάτων, καθώς και για το ενδιαφέρον του καθόλη τη διάρκεια της εργασίας διακρίνοντας τον πάντα εξαιρετική ευγένεια και αμεσότητα.

Τον κ. Τσαμαρδίνο για τη συμβολή του στη στατιστική ανάλυση των δεδομένων, τις καθοριστικές παρεμβάσεις του, τη συνεισφορά του σε κάθε φάση αυτής της μελέτης πάντοτε με μεγάλο ενθουσιασμό και φιλόδοξη οπτική.

Καθώς οι ευχαριστίες δεν αρέσουν στον Ανδρέα, θα του υπενθυμίσω μόνο το πόσο πολύτιμος είναι στη ζωή μου. Δε θα ξεχάσω τις ατελείωτες συζητήσεις μας, την υπομονή και την κατανόηση του σε όλες τις στιγμές του παραλογισμού μου αλλά και την αγάπη του όλα αυτά τα χρόνια. Ο Ανδρέας αποτελεί το καταφύγιο μου, την ουσία μου [1] και μαζί με τους φίλους μας Μάνθο, Μανόλη και Χριστίνα θωρακίζουν την καθημερινότητα μου και δίνουν αξία στις στιγμές. Είναι υπέροχο να μεγαλώνουμε μαζί και η φιλία μας είναι η σπουδαιότερη επιτυχία μου μέχρι τώρα.

Ευχαριστώ τα κορίτσια Δήμητρα, Ειρήνη, Μαίρη και Νέλη για την καθημερινότητα που μοιραστήκαμε, τις κοινές ανησυχίες, τα μελλοντικά σχέδια που σημάδεψαν τα φοιτητικά μας χρόνια και αποτέλεσαν την αφορμή μιας δυνατής φιλίας.

Τέλος, δε μπορώ να παραλείψω τους γονείς μου Κάτια, Γιάννη για την αγάπη και την ενθάρρυνση τους σε όλα τα χρόνια των σπουδών μου, το Νίκο και την αναντικατάστατη αδερφή μου Φωτεινή που με μια της κουβέντα μπορεί να απλοποιήσει όσα στο δικό μου μυαλό φαντάζουν σύνθετα.

[1] '..η ενυπάρχουσα αρχή και αιτία της υπάρξεως όντος..' κατά Αριστοτέλη.

Contents

1 Introduction **1**
- 1.1 Motivation . . . 2
- 1.2 Problem Statement & Contributions . . . 4
- 1.3 Thesis Organization . . . 8

2 Preliminaries on Text Analytics **9**
- 2.1 Text Analysis . . . 11
  - 2.1.1 Lexical Analysis . . . 11
  - 2.1.2 Information Extraction Techniques . . . 12
  - 2.1.3 Information Weighting Schemes . . . 13
- 2.2 Text Clustering Algorithms . . . 15
  - 2.2.1 Partitioning Algorithms . . . 16
  - 2.2.2 Hierarchical Algorithms . . . 17
  - 2.2.3 Density-Based Algorithms . . . 17

3 Social Media **21**
- 3.1 Overview . . . 21
- 3.2 Online Social Networks . . . 22
- 3.3 Twitter Microblogging Framework . . . 24
  - 3.3.1 Twitter Information Model . . . 25
  - 3.3.2 System Architecture . . . 27
  - 3.3.3 Statistical Analysis . . . 30

4 Related Work on Clustering Streams **39**
- 4.1 Stream Clustering Algorithms . . . 39
- 4.2 Social Stream Mining . . . 43
  - 4.2.1 Challenging Issues . . . 44
  - 4.2.2 Clustering Algorithms . . . 44

5 A Realistic Workbench **49**
- 5.1 Datasets . . . 49
- 5.2 Sampling Methodology . . . 51

6 Statistical Analysis of Social Text **55**
6.1 Clustering Parameters 56
6.2 Cluster's Properties 60
6.2.1 Centroid Trajectory 60
6.2.2 Shape Evolution 72
6.2.3 Shape Recognition 77

7 Conclusions & Future Work **81**

List of Figures

1.1 Illustration of the k-means clustering algorithm's behaviour 3
1.2 Entropy for k-means & TStream algorithms 6

2.1 Main stages of the clustering procedure 10

3.1 Map of social networks' popularity 23
3.2 Twitter information model 26
3.3 JSON representation of a tweet 26
3.4 System architecture 27
3.5 Sub-components of the core component 28
3.6 An Entity-Relationship diagram of Twitter 29
3.7 Acquisition percentage per month 30
3.8 Percentage of new accounts per year 31
3.9 (a) Twitter users language (b) Twitter users time zone 31
3.10 Top third-party applications 32
3.11 Granularity level of place 32
3.12 Number of users with X followings 33
3.13 Number of users with X followers 34
3.14 Correlation between followers and followings 35
3.15 Number of users with X posts 36
3.16 Number of tweets against the number of followers 37

4.1 Architecture of a stream clustering system 41

6.1 Proximity and ground truth matrices 56
6.2 Comparison of weighting schemes & distance metrics, sample III . 58
6.3 Comparison of weighting schemes, sample III 59
6.4 A continuous post stream using count-based sliding window 61
6.5 Hotelling's T-squared test, sample I, topic: 'Libya' 62
6.6 Hotelling's T-squared test, sample II, topic: 'Libya' 62
6.7 Hotelling's T-squared test, sample III, topic: 'Japan' 63
6.8 Centroid trajectory in 3D of the three samples 64
6.9 Hotelling's T-squared test over time windows, sample I 66
6.10 Hotelling's T-squared test over time windows, sample II 67
6.11 Hotelling's T-squared test over time windows, sample III 68

6.12 Histogram of distances from cluster's centroid over time 72
6.13 Variance of distances from cluster's centroid, constant FS 77
6.14 Variance of distances from cluster's centroid, non constant FS . . . 78
6.15 Points variance from centroid for constant and non-constant feature space 79

List of Tables

1.1 K-means quality results & top words per cluster 3
1.2 TStream quality results & top words per cluster 4
1.3 Parameters of TStream algorithm 7

3.1 Basic characteristics of the top 3 social media 24
3.2 Basic statistics over our Twitter dataset 30
3.3 (a) Top cited countries, (b) Top cited neighbourhoods 33
3.4 Type of tweets 37

5.1 Basic characteristics of dataset I 50
5.2 Basic characteristics of dataset II 50
5.3 Distribution shape over time & characteristics of sample I 51
5.4 Distribution shape over time & characteristics of sample II 52
5.5 Distribution shape over time & characteristics of sample III 53

6.1 Table of symbols 57
6.2 Hotelling p-values for different values of n_{step} 64
6.3 Hotelling p-values for different values of t_{step} 69
6.4 Hotelling p-values for different values of t_{step} for bursT method . . 71
6.5 Kolmogorov-Smirnov test results for the count-based window model 73
6.6 Kolmogorov-Smirnov p-values for different values of n_{step} 74
6.7 Kolmogorov-Smirnov test results for the time-based window model 75
6.8 Kolmogorov-Smirnov p-values for different values of t_{step} 75
6.9 Kolmogorov-Smirnov p-values for different values of t_{step} for bursT method 76

Chapter 1

Introduction

During the last decades we witnessed of a continuously increasing rate of publishing data on the web. The official news sites as well as the social media have a great contribution on data sharing. The ease of publishing, searching and accessing information on the web encourages the individual users to communicate their content with the web society. In addition, the social media made the distribution of User-Generated Content (UGC) more interactive and valuable. Users are able not only to post their opinion, but to comment on others' people text. Each posted text is enriched with meta-data revealing the identity of the author, the location or posting time etc.

Online networking sites (e.g. Facebook), micro-bloging applications (e.g. Twitter) and Social news (e.g. Digg), apart from boosting the web usage by individuals, also produce on a daily basis vast amount of UGC related to a wide variety of real-world news (personal, political, commercial etc). The real-time communication is not only a privilege or a right, but also a tool in critical historical moments. It is worth mentioning that during the events in Libya (17 February 2011 – 23 October 2011) two revolutions took place. The first was given in the streets of Libya and the second in the virtual community of Twitter. Both of them claimed and marked the beginning of a new era in the political development of the local society, as well as in the characterization of the social networks as fertile sources of news. Twitter was ahead of news-wire, spreading the news and public opinion to the official news media. The textual information and the attached meta-data make the need of processing the upcoming information imperative and challenging.

The massive stream of digital text has attracted the attention of marketers, politicians and scientists and leads to the need of automated analysis. By analysing the stream of communications in an unmediated way many scientists are having direct access to people's opinions and observations for the first time.

Several machine learning clustering methods have been proposed in this respect during the last years. However, there is still no commonly used methodology for statistical analysis of textual content produced in social media that take into account the peculiarities of social text streams exhibited in reality.

For instance, Twitter posts are *short* texts of up to 140 characters with lots of spam and *low quality* text. Those characteristics affect the extracted vocabulary and its weighting representation. Furthermore, the content of Twitter posts is heterogeneous and noisy ranging from breaking news to personal stories affecting the number and utility of the recognised clusters. Finally, tweets are highly evolving characterized by a non-stationary data distribution and clusters of varying volume, shape and density.

1.1 Motivation

Towards the first steps of this thesis we performed an experimental study utilizing the k-means traditional algorithm proposed in literature, as well as a more sophisticated method for stream-oriented data (e.g. TStream) in order to understand if the existing algorithms can be adapted to the peculiarities of the social text stream already mentioned.

Our set of experiments is applied over three different samples capturing various aspects of the social stream nature. The first sample consists of topics occurring sequentially over time collected from Twitter under the tags '*Libya*' and '*champions league*'. The second sample is collected around the bursty days of the topics '*flotilla*', '*Libya*' and '*champions league*', while the last one encompasses topics happening simultaneously in time ('*Libya*', '*champions league*' and '*Japan*'). A detailed discussion of the samples can be found in Chapter 5.

The experimental results for the k-means [1] algorithm, performed under ten repetitions, reveals a poor clustering quality measured with average precision and NMI metrics. The Table 1.1 depicts the top five words of each cluster for the pre-defined k value of every sample. The algorithm fails to distinguish the different topics resulting in misclassified tweets and many clusters of the same topic area. Also, it appears weak in capturing the heterogeneity of topics existing in the samples although it has the knowledge of the whole dataset a-priori as well as the number of the expected clusters. The low performance arises questions on the scalability of the algorithm in real stream conditions.

To illustrate the behaviour of the algorithm we can imagine tweets as colourful messages where messages of the same color need to be grouped together. K-means algorithm, depicted in Figure 1.1, creates clusters with the same prevailing topic but having misclassified several messages.

The last algorithm, TStream [2], belongs to the category of hierarchical algorithms that use sliding windows to deal with the stream. It organizes the stream into two-level hierarchy of broad topics and more specific subtopics. We performed our experiments using the implementation of the algorithm provided by the authors. In order to obtain comparable results with the previous algorithms we set the size of the window equals with the size of the sample. The initialization of the several different parameters of the algorithm makes its usage difficult and oriented to the particular dataset. After several combinations, but certainly not exhaustive,

	Cluster I	*Cluster II*	*Cluster III*	*Avg. Precision / NMI*
Sample I (k=2)	*mad* *libyan* *gaddaficrim* *tripolitan* *lif*	*libyan* *gaddaficrim* *arab* *aljazeer* *tripolitan*		48.75% / 0.17%
Sample II (k=3)	*tripolitan* *gadhaf* *aljazeer* *arab* *head*	*libyan* *mad* *arab* *million* *aljazeer*	*leav* *championsleagu* *fir* *bas* *upd*	66.48% / 51.17%
Sample III (k=3)	*japan* *east* *tumblr* *du* *nuk*	*libyan* *mad* *alread* *japan* *arab*	*mad* *japan* *east* *tumblr* *nuk*	64.76% / 56.48%

Table 1.1: K-means quality results & top words per cluster

we ended up with the results presented in Table 1.2. The top words per cluster and sub-clusters are presented in the Table 1.2 indicating various sub-topics discussed by the Twitter users under a general thematic area. For instance, some of the revealed sub-topics are the rumour that Satoshi Tajiri, the creator of Pokemon, has died in Japan's earthquake or the Twitter users' appeal for donation to the victims of the earthquake.

Motivated by the aforementioned results we try to better understand the peculiarities of the social content as well as the drawbacks of the clustering algorithms and define the borders where each of them affect the clustering quality. In the next section, we study the k-means and TStream algorithms under ideal experimental conditions, we define the problem statement of the thesis and extract valuable observations.

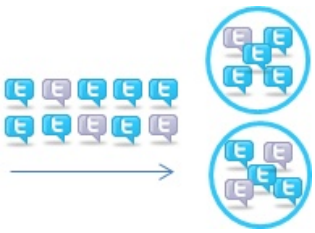


Figure 1.1: Illustration of the k-means clustering algorithm's behaviour

Av. Precision / NMI	*Cluster I*	*Sub-clusters*	*Cluster II*	*Sub-clusters*	*Cluster III*	*Sub-clusters*
I 50%/ 1.6%	*rt*	*rt*	*tripoli*	*tripoli*		
	u	*al*	*rt*	*rt*		
	oil	*libia*	*people*	*protesters*		
	s	*protesters*	*now*	*gadafi*		
	sanctioned	*egypt*	*reports*	*people*		
	protesters	*u*	*libyan*	*rt*		
	libia	*oil*	*more*	*now*		
	obama	*s*	*rights*	*people*		
	egypt	*sanctioned*	*protesters*	*tripoli*		
	al	*rt*	*egypt*	*libyan*		
II 33.4%/ 5.3%	*french*	*french*	*benghazi*	*action*	*u*	*u*
	military	*first*	*rt*	*military*	*s*	*s*
	fire	*operation*	*action*	*going*	*missiles*	*war*
	jet	*fire*	*forces*	*forces*	*rt*	*prepare*
	plane	*rt*	*news*	*rt*	*obama*	*against*
	rt	*jet*	*over*	*benghazi*	*prepare*	*missiles*
	vehicle	*over*	*world*	*rt*	*war*	*now*
	first	*fly*	*going*	*short*	*against*	*u*
	targets	*french*	*tripoli*	*news*	*launch*	*s*
	operation	*fighter*	*now*	*nabbous*	*cruis*	*launch*
III 46.3%/ 3.3%	*nuclear*	*tsunami*	*01*	*11*	*libya*	*earthquake*
	rt	*today*	*quake*	*live*	*earthquake*	*tsunami*
	plant	*died*	*up*	*lost*	*pray*	*rt*
	libya	*satoshi*	*victims*	*12*	*rt*	donated
	tsunami	*creator*	*retweet*	*pleas*	*tsunami*	*help*
	news	*libya*	*give*	*01*	*go*	*libya*
	tripoli	*nuclear*	*rt*	*quake*	*help*	*pray*
	today	*rt*	*more*	*up*	*u*	*rt*
	died	*plant*	*donated*	*victims*	*out*	*people*
	now	*news*		*retweet*	*people*	*u*

Table 1.2: TStream quality results & top words per cluster

1.2 Problem Statement & Contributions

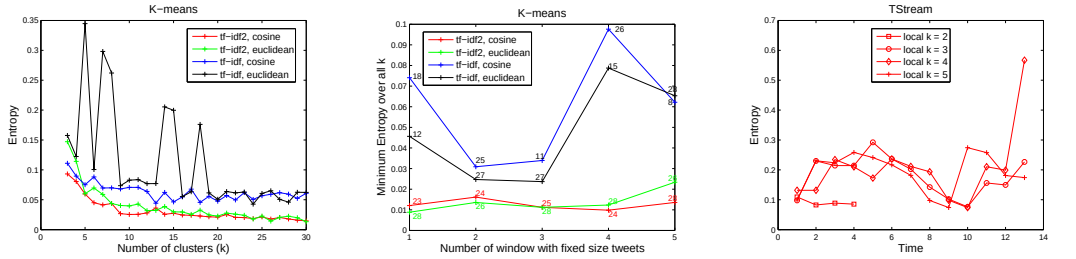
We are focusing on the clustering of tweets based on their textual content since it is the most informative part of social messages and the #tags under which they are posted to index them in high-level thematic categories. Compared to these categories, the detected clusters are expected to capture more fine-grained topics

(i.e. sub-topics) of the conversation conducted in a social stream. To this end, we use two state-of-the-art clustering algorithms, namely k-means [1] and TStream [2], over a sample of 10000 tweets from our workbench which cover three thematically independent #tags occurring simultaneously in time with much different arrival rates and volumes. These algorithms aim to partition incoming tweets to a fixed number k of (sub-)topics. As we will see in the sequel the best possible k w.r.t. the resulting clustering quality strongly depends on the calibration parameters of the two algorithms for our sample dataset. As quality criterion we have chosen the conditional entropy [3] of the #tag of a tweet given its cluster $H(Tag \mid C)$ with the following reasoning: a low $H(Tag \mid C)$ implies that knowing the cluster of a tweet, there is little uncertainty about its #tag. Thus, this metric does not penalize for a clustering that partition the tweets published under the same #tag to several clusters, corresponding to multiple sub-topics, but penalizes only when tweets from different #tags are placed in the same cluster.

K-means is based on an iterative partitioning of the input points into k clusters, in the aim to minimize the sum, over all clusters, of the within-cluster sums of *point to cluster centroid distances*. To estimate the number of clusters resulting to a minimal entropy, we evaluated the algorithm for several k-values carrying out 20 iterations over our sample dataset. Since the within-cluster distance is affected both by the employed weighting scheme (WS) of terms and the similarity metric, in Figure 1.2(a) we report several alternative choices, among which $tf \cdot idf_2$[1] and cosine similarity [5] exhibit for our dataset the best possible clustering quality (i.e. the lowest entropy). K-means fails to detect clusters for the #tag with the smaller number of tweets that it just constitutes the 0.03% of the total dataset. It worth mentioning that in this experiment the entropy does not monotonically drop as the number of clusters increases. That is due to the way entropy is computed: it is the estimated rather than the true entropy since the true probabilities of the population distribution are unknown. Furthermore, k-means is a heuristic algorithm which does not guarantee an optimal clustering. As a result in the estimated entropy calculations a variance is introduced that increases with k, causing statistical fluctuations. K-means quality is substantially affected by the number of dimensions of the Feature Space (FS) employed to compute the distance of points. The entropy reported in Figure 1.2(a) has been computed for a constant FS with the 500 most frequent occurring terms in the entire sample dataset after ignoring stopwords. Since the vocabulary of tweets actually evolves over time along with the drift in user conversations, we need to dynamically adjust not only the number of detected clusters but also the FS on which they are computed.

To this end, Figure 1.2(b) depicts the entropy (y-axis) when k-means is independently executed on consecutive, non-overlapping time-windows with 2000 tweets

[1] $tf \cdot idf_2$ differentiates from the well-known $tf \cdot idf$ weighting scheme only in the first factor tf that refers to the word's frequency occurrence in the entire corpus instead of the single tweet. Some related works [4] use $df \cdot idf$ WS instead, where df counts the number of tweets containing a word. However multiple appearances of the same word in a tweet are rare making the two WS similar.



(a) K-means results for static analysis (constant FS) of different WS: $tf \cdot idf_2$ & cosine robust in time

(b) K-means results for dynamic analysis (adjustable FS) of different WS: $tf \cdot idf_2$ & cosine robust in time

(c) Clusterings of TStream in different windows are related: calibration of parameters is a critical task

Figure 1.2: Entropy for k-means & TStream algorithms

each (x-axis) by exploring an adjustable FS with the most frequent 500 terms occurring in each window[2]. For all possible combinations of terms' weighting schemes and similarity metrics we report above the four curves the number of clusters for which the smallest estimated entropy is observed in each window. The fluctuation of entropy for the two $tf \cdot idf$ curves is more important than for $tf \cdot idf_2$ while the absolute entropy values computed in the last window of 2000 tweets are very close to (vs. considerably diverge from) the values of the latter (vs. former) computed in the entire dataset of 10000 tweets (see Figure 1.2(a)). Note that in windows 3, 4 $tf \cdot idf_2$ curves seem to be also resistant to bursty arrival of tweets (from #tag Japan). It should be however stressed that k-means produces un-nested and non-overlapping spherical clusters without any information regarding the relationship among them. In addition the independent execution of k-means in each window completely forgets past tweets in previous ones making difficult to understand the evolving shape and density of the detected clusters across windows.

For this reason we have additionally considered the hierarchical clustering algorithm TStream, which detects spherical clusters of high-similarity (i.e. sub-topics) nested within wider ones (i.e. topics). In order to update such a two-level clusters hierarchy when a number of novel data is detected, TStream periodically reorganizes either the first or the second level of clusters by recomputing their FS based on the new collection and the memory of the W latest data ($tf \cdot idf_2$ weighting [3]). Table 1.3 presents a high number of parameters that need to be initialized in advance for the TStream algorithm. A first effort towards the parameters calibration has been made based on the analysis of Section 6 and the knowledge of k-means results. The *global* and *local container* (GC/LC) has been set to 600 since this number of novel tweets has been proven statistically significant in our dataset (see Section 6) to trigger a first or second level re-clustering (i.e. topic evolution).

[2]We observed that increasing the dimensions over 500 has no significant benefit to clustering, while fewer than 500 dimensions result to too many zero weighting vectors.

[3]For consistency reasons, we changed the $tf \cdot idf$ WS used by TStream into $tf \cdot idf_2$.

The GC cosine similarity threshold was set to the average similarity of k-means clusters (cosine, $tf \cdot idf_2$) containing tweets with the same #tag for which the lowest entropy ($k = 30$) is exhibited while the LC similarity threshold was defined as the minimum cosine metric detected in the clusters. The number of the latest tweets (W) considered in case of re-clustering was empirically set to the size of two windows. The *kglobal* parameter refers to the number of the first-level clusters capturing distinct #tags, whereas *localk* defines the number of second-level clusters where at least two sub-topics were detected by k-means for each #tag. We used 2000 tweets as the initialization step (*initialDocNo*) of the algorithm and tumbling windows (Wsz) of the same size. The dimensions of clusters' centroid (*centroidSz*) as well as the size of the input word vectors ($WordSz$) were set to 500.

global σ (GC)	**local σ (LC)**	**global δ (GSim)**	**local δ (LSim)**	**W**	**WSz**
600	600	0.52	0.7	2	2000

global k	**local k**	**initialDocNo**	**centroidSz**	**WordSz**
2	$[2, 5]$	2000	500	500

Table 1.3: Parameters of TStream algorithm

Figure 1.2(c) depicts the entropy for the top-level clusters detected by TStream for various values of *localk*. The entropy of TStream clustering appear to be in most cases at least one order of magnitude higher than k-means. Unlike Figure 1.2(b), where the x-axis refers to sequential fix-sized windows, in Figure 1.2(c) the x-axis refers to the relative time points in which different volumes of tweets are clustered. Thus, the various curves of the entropy are not directly comparable. We can observe downward or upward trends in the entropy caused by the global, local or no re-clustering decisions of TStream depending on the actual parameters' value and the tweets of each time window. For instance, for *localk*=2 no re-clusterings are performed as there are not enough *novel* tweets above the novelty thresholds (GC/LC) to trigger such process. Since neither the global nor the local hierarchy is re-organized, the new incoming tweets are clustered into the already existing clusters and thus the curve (*localk*=2) shows the entropy at the time points where a new window ($Wsz = 2000$) is added. We can observe that this is the only case where TStream results to the same clustering quality as k-means for k=4. In particular, TStream maintains two top-level clusters corresponding to the high-level topics 'Japan' and 'Libya' each one containing second-level clusters with tweets sharing the same #tag, while k-means results to three clusters containing tweets from the #tag 'Japan' and one from 'Libya' with only few miss-classifications (<300 tweets).

In both algorithms, to improve the clustering quality a thorough calibration of their input parameters is needed which implies to understand how core cluster

properties evolve over time. As we have seen a *static* parameters calibration is not always able to improve the clustering quality. For this reason, we propose an original methodological framework for recognizing how clusters *centroid*, *shape* and *density* evolve along with the *granularity* and *dynamics* of user conversations in real tweets. We believe that this framework is essential in order to enable a *dynamic* adaptation of the parameters impacting the quality of the clustering results achieved by different algorithms.

1.3 Thesis Organization

After this introductory section, in Chapter 2 we define the preliminaries of Social Analytics. This includes some basic notions from lexical analysis, information extraction and weighting schemes as well as some fundamental clustering algorithms. In the Chapter 3 we refer to the system's architecture used to collect and analyse Twitter data and provide the reader with a characterization of Twitter social network based on a statistical analysis. Chapter 4 describes the related work on clustering streams of documents and social stream mining. In Chapter 5 we present a realistic workbench that takes into account the peculiarities of social text streams exhibited in reality. Chapter 6 builds on the the previously presented framework, in order to study the clustering parameters and capture the evolution of the clusters properties. We conclude this thesis in Chapter 7 referring also to possible future work.

Chapter 2

Preliminaries on Text Analytics

In this chapter we go through some fundamental theoretical notions which serve as a basis for our work. We start by defining the general field of this research project, that is the task of text clustering [6]. We move on by presenting some basic methods of text mining including lexical analysis, information extraction and weighting schemes. Then, we present three traditional, well-known in literature, clustering techniques.

Clustering is a widely studied data mining problem in a variety of fields. The task of clustering assigns points in a multidimensional space into cohesive groups of similar objects, called clusters. The similarity between the data points is measured with the use of a similarity function. Each cluster consists of points that are similar between themselves and dissimilar to points of other groups. The problem of clustering is especially useful in the text domain and News Event Detection (NED) and it was early studied in literature.

The first systematic work concerning document clustering has been done during the Topic Detection and Tracking (TDT) research initiative, where a number of algorithms have been proposed solving the problem of detecting *topics* and *events* in a corpus of documents. The distinction of these terms is mentioned early in the literature and several paradigms were given. Among them, an "Earthquake" can be considered as a topic but "Earthquake in Japan, 11/03/2011" concerns an event belonging to the general topic discussing earthquakes. The notion of an event differs from that of the topic in spatial and temporal localization and in specificity. Several times the distinction is rather difficult even in a conceptual mode, making clear the difficulties of the algorithms. Several methods were applied in order to refine the clustering results and to achieve high clustering quality.

Topic Detection and Tracking (TDT) aims at finding techniques that group news documents into event documents. Formally, the topic in TDT is defined as a seminal event or activity, along with all directly related events and activities. An event is defined as something that happens at some specific time and place along with all necessary preconditions and unavoidable consequences. Story is a topically cohesive segment of news that includes two or more declarative independent clauses

about a single event.

The TDT task can be distinguished and simulated by the following five research and development challenges.

1. Story Segmentation - The main goal of this task is to reveal from a stream of documents, those belonging to the same story.

2. First Story Detection / News Event Detection (2002) - First Story Detection is the task of deciding if a new story of the stream is discussing a new topic.

3. Cluster Detection - Cluster detection is the problem of grouping together stories that discuss the same topic. As a story arrives, it is grouped to the relative cluster.

4. Tracking - Tracking is the problem of finding all relevant stories of a predefined topic.

5. Link Detection - Link detection refers to the problem of deciding if two stories are associating. More specifically, if they discuss the same topic.

In the field of NED an important boost in research was given by the TDT research project and a remarkable progress was made. One common approach in NED task is the use of machine learning algorithms. Algorithms of this category deal with the problem of NED as a clustering problem, where each cluster refers to a specific event and different clusters correspond to different events.

Three are the main stages of the clustering procedure (Figure 2.1). The first refers to the text analysis, where the most meaningful and representative terms are extracted from the text. The indexed terms are being weighted forming a Vector Space Model (VSM). The second stage exploits the produced VSM by grouping the documents based on their computed similarity. Last but not least, the third stage evaluates the clustering procedure using supervised (for a known output) and unsupervised (for unknown output) quality measures. In the next section, we discuss the first two stages, by presenting the steps of text analysis and reviewing three traditional well-known related techniques on clustering.

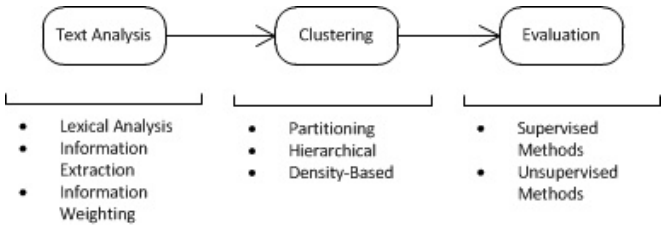


Figure 2.1: Main stages of the clustering procedure

2.1 Text Analysis

The field of Text Analysis, also mentioned as Text Analytics in business applications, models and structures the information content of textual sources (i.e documents) with the ultimate goal to explore, analyse and extract intelligence from data. The textual content may be derived from different sources, like businesses, networks, social media etc. The analysis of this content, which is massive in most of the cases, is closely related with techniques of lexical analysis, term extraction and terms' weighting schemes (later discussed in this section). The overarching goal of text analysis is, essentially, to turn text into data for analysis via application of natural language processing (NLP) and analytical methods. The first step towards this goal is the extraction of the most representative words, the reduction of the data dimensions and then the weighting of the selected terms.

2.1.1 Lexical Analysis

The preprocessing procedure of lexical analysis has a great impact on the quality results of the clustering. It consists of five main steps where plain text is given as input and a vector of tokens is returned as output. Each of that step is described shortly below and has been applied to the majority of text mining works.

- **Filtering** The filtering process removes unnecessary tags and extracts useful information under some specified patterns. For example, consider pulling out the links of a web page by removing the unnecessary tags of the HTML file.
- **Tokenization** The tokenization step splits the sentences into tokens, typically words. In some cases, more sophisticated techniques are applied, such as grammatical structure extraction.
- **Stemming** The stemming step converts the words to their stem, base or root form. The stem need not be identical to the morphological root of the word; it is usually sufficient that related words map to the same stem. Algorithms for stemming have been studied in computer science, with the most well known among them the Lovins stemmer [7] and especially for the English language the Porter's stemmer [8] algorithm.
- **Stopword Removal** A stopword is defined as a term that is commonly used and is met very frequently in texts. In order to save both space and time, these words are dropped at indexing time. A typical method to remove stopwords is by comparing the terms with a known stoplist.
- **Pruning** The last step of pruning usually removes words with very low frequency throughout the corpus. The underlying assumption is that these words correspond to low-content and have low discriminating power. On this basis, the hypothesis is that pruning the words corresponding to low-frequency and thus to content-pure words, may enhance lexical analysis performance [9].

2.1.2 Information Extraction Techniques

Information extraction step is of a great importance to the clustering process, as the quality of the clustering is highly dependent on the appropriateness of the input features. In some cases, where the data are described with a large number of features, it is beneficial to reduce the dimensions of the input. Feature selection and feature extraction are the two major techniques for reducing the dimensionality of the input data and provide better text representation.

1. **Feature Selection**

   Feature selection technique locates the best minimum subset of the original features in order to reduce the initial dimensions. For the purpose of text clustering, the objective of feature selection is to identify features (corresponding to words) that predict the class label and construct accurate clusters. Feature selection strategies can generally be divided into two broad categories, filter and wrapper.

   - Filter Approach
     Filter approach selects the most relevant attributes and removes the irrelevant features prior to the clustering process. For example, the most common and simple practice for reducing the features of a given collection is the removal of stop-words, a set of frequently met words such as "the", "is", etc. These words are noisy, with very low discriminating capability and thus they complicate the clustering technique. In the field of filtering, the utility of named entities has been examined [10, 11] and some named entities (i.e location, organization, date, time) have been re-weighted to contribute to the similarity of the documents [12]. There are acknowledged conclusions [13] that in some cases the usage of named entities is significant to clustering quality.

   - Wrapper Approach
     Wrapper approach is a greedy searching method, which divides the feature space of words into subsets and attempts to find the optimal subset that maximizes a given metric. Specifically, the method iteratively evaluates a candidate subset, then modifies the subset and evaluates if the new subset is an improvement over the old. Evaluation of the subsets requires a scoring metric that grades a subset of features. Exhaustive search is generally impractical, so a stopping criterion is defined; possible criteria include: a subset score exceeds a threshold, a program's maximum allowed run time has been surpassed, etc.
     A hybrid algorithm applying both filter and wrapper methods with an expectation maximization criterion is applied in [14] producing promising quality results in terms of accuracy and Normalized Mutual Information.

2. **Feature Extraction**

   Feature extraction technique transforms the data of a high-dimensional space into a space of fewer dimensions. It produces a new set of features from the original data through some mapping. The most well-known unsupervised feature extraction method is the Principal Component Analysis (PCA). PCA is unsupervised as it does not take into account the class labels. It performs a linear mapping of the data to a lower dimensional space in such a way that the variance of the data in the low-dimensional representation is maximized. In addition, it offers a convenient way to control a trade-off between loosing information and simplifying the problem of dimensionality. Most of the modern methods for nonlinear dimensionality reduction find their theoretical and algorithmic roots in PCA and they have been successfully applied in the field of document clustering [15]. An extended survey of feature reduction techniques can be found in [16, 17].

2.1.3 Information Weighting Schemes

The last step of pre-processing text analysis, where the input data are being prepared to feed the clustering algorithm, is the weighting of the extracted tokens. Many schemes have been proposed [18, 19, 20, 21] emphasizing in different aspects of textual data.

A common weighting scheme for terms within a document, known from the field of information retrieval and commonly used, is the Vector Space Model (VSM). VSM is an algebraic model for representing objects as vector of identifiers. In the concept of NED, an object is a document, where each dimension corresponds to a weighted term (i.e single word). The prevailing technique of term weighting is the TF-IDF scheme. In the basic model, the TF component represents the frequency of a term w in a specific document d and IDF is intended to discount very common words in the collection. Below is the particular tf-idf scheme used in [18].

$$weight(w, d) = tf \cdot idf \tag{2.1}$$

$$tf = log(\, termfrequency + 1 \,) \tag{2.2}$$

$$idf = log(\, \frac{docCount + 1}{documentFreq + 0.5} \,) \tag{2.3}$$

where $docCount$ is the number of documents in the corpus and $documentFreq$ is the number of documents containing the term t.

Many variations of the above scheme have been proposed, based on the observation that the corpus is not always known a priori. The authors in [19] extend the basic model to an incremental one, where the document frequencies are not static but change dynamically in time (equation 2.4).

$$df_t(w) = df_{t-1}(w) + df_{C_t}(w) \tag{2.4}$$

For each set of documents C_t added in the corpus during the time interval t, the document frequencies (df) and inverse document frequencies (IDF) are updated based on the new batch of documents. Additionally, the terms with low frequency (below a certain pre-defined threshold) are ignored as uninformative. Finally, the weight of the word w in the document d at time t is defined as:

$$weight_t(d,w) = \frac{1}{Z_t(d)} f(d,w) \cdot log \frac{N_t}{df_t(w)} \tag{2.5}$$

where N_t is the total number of documents at time t, $f(d,w)$ is the frequency of word w in document d and $Z_t(d) = \sum_w f(d,w) \cdot log \frac{N_t}{df_t(w)}$ is a normalization value.

In [20] the authors introduce the use of a time window, containing a fixed number of documents. Each time a new document is processed the idf term is recomputed and the vocabulary is updated. The particular incremental idf metric is defined to be:

$$idf = log_2(\frac{N_{(p)}}{n_{(t,p)}}) \tag{2.6}$$

where p is the current time, t is a term, $N_{(p)}$ is the number of documents up to the current point and $n_{(t,p)}$ is the number of documents which contain term t up to the current point p.

More recent works are dealing with the challenging issue of mining real-time content in a dynamic environment. The authors of [21] suggest a weighting method, called BursT, that uses sliding windows and takes into consideration two factors: Burst Score (BS) and Term Occurrence Probability (TOP).

$$weight_{w,t} = BS_{w,t} \cdot TOP_{w,t} \tag{2.7}$$

The burst score (equation 2.8) is the deviation between the arrival rate of the word w at time t ($ar_{w,t}$) and its expectation value ($E(ar_{w,t})$). Symbol $at_{w,t}$ denotes the arrival time of the word w occurred in the t-th arrival message of the messages sequence.

$$BS_{w,t} = max\{\frac{ar_{w,t} - E(ar_{w,t})}{E(ar_{w,t})}, 0\} \tag{2.8}$$

$$ar_{w,t} = \frac{1}{at_{w,t} - at_{w,t-1} + 1} \tag{2.9}$$

The TOP factor (equation 2.10) denotes the term occurrence probability corresponding to the word w at t-th arrival, where C_t is the message collection in the corpus collected from the time t-tw to current time.

$$TOP_{w,t} = P(w_t|C_t) = \frac{|m : w_t \epsilon C_t|}{|C_t|} \tag{2.10}$$

2.2 Text Clustering Algorithms

The text-based document clustering algorithms characterise each document according to its content, i.e. the words (or sometimes phrases) contained in it. The basic idea is that if two documents contain many common words then it is likely that the two documents are very similar. The similarity of the documents with the existing clusters is computed using distance functions and similarity metrics. In their majority traditional methods require an a-priori knowledge of the number of clusters in order to decide the target cluster of each new document. Such algorithms, like the K-means and EM [22], aim to minimize the within-cluster distance. Each cluster is represented by the mean (or weighted average) of its points, the so-called centroid and euclidean or Hellinger [19] distance is usually computed. Other approaches, such as the threshold-based algorithm, use similarity metrics. Many similarity metrics have been proposed [19] with the most popular among them, the cosine similarity [10]. The similarity between document d and cluster c can be calculated as in equation 2.11.

$$sim(d,c) = \frac{\sum_w weight(w,d) \cdot weight(w,c)}{\sqrt{\sum_w weight(w,d)^2}\sqrt{\sum_w weight(w,c)^2}} \tag{2.11}$$

Where each word w is included in the weighting vector of the document d and cluster c. The cosine treats both vectors as unit vectors by normalizing them, giving a measure of the angle between the two vectors. It does provide an accurate measure of similarity but with no regard to magnitude (i.e. if a word occurs in both vectors - no matter the number of occurrences - they hint at the same direction). If magnitude (euclidean distance) was taken into account, the results would be quite different.

Based on cosine similarity, the decision of whether the input document belongs to an existing cluster is taken by the use of a pre-defined threshold parameter. If the similarity is below a certain threshold the document is considered to discuss a new topic and is mapped to a new cluster. Otherwise, the document is grouped to an existing cluster with the highest similarity score. Many improvements of that technique have been applied. Some of them can be found in the survey of clustering data mining techniques [22].

The threshold-based algorithm was invented by the TDT team at the Carnegie Mellon University (CMU) to decide whether a new document is another story of one of the detected events or it belongs to a new event of its own. The simplest form of the proposed algorithm, which relies only on pair-wise document similarity information is portrayed in Algorithm 1.

Algorithm 1 $FSD(d, [c_1, ..., c_k, ...])$

Require: d: the current incoming document
Require: $[c_1, ..., c_k, ...]$: the existed clusters

for all clusters $c_j \in [c_1, ...c_k, ...]$ **do**
    $sim_{d,c_j} \leftarrow sim(d, c_j)$
    if $sim_{d,c_j} > maxsim_{d,c_j}$ **then**
        $maxsim_{d,c_j} \leftarrow sim_{d,c_j}$
    end if
end for
if $maxsim_{d,c_j} > t_c$ **then**
    merge(d, c_j)
else
    if $maxsim_{d,c_j} > t_n$ **then**
        new cluster containing d
    end if
end if

Similarity functions can be used in conjunction with a wide variety of traditional clustering techniques, the most prominent among them are partitioning, hierarchical and density-based algorithms. These techniques will be discussed in the next subsections.

2.2.1 Partitioning Algorithms

The partitioning methods divide the initial dataset D into sets of k clusters, where each object belongs to only one cluster. They create a one-level, un-nested partitioning of the data points where the number of clusters is given as input. Each cluster is described by a centroid or a representative, which is a summary of the objects. The precise form of the centroid depends on the type of the clustered objects. In case of real-valued data the arithmetic mean of each attribute is calculated. In case of documents the mean value is calculated based on the weighting metric.

There are a number of partitioning techniques, but we shall only describe some of them applied in the field of document clustering.

The most representative algorithm of this category is k-means [1]. The basic k-means algorithm follows the concept of centroid to represent the cluster. At the beginning, it selects K points as the initial centroids and it assigns all points to their closest centroid. Then, it recomputes the centroid of each cluster and repeats the same procedure until it converges and there are no changes to the centroids.

Fuzzy c-means [23] is another well-known unsupervised clustering technique that it also requires the number of clusters as input. It differs from k-means as it calculates the document's degree of membership to every existing cluster.

An experimental study on text documents [24] showed that fuzzy clustering is

a more stable method, as it produces better results on almost all datasets. Some other studies [25] compare these algorithms with hierarchical approaches discussed in the next section.

2.2.2 Hierarchical Algorithms

The hierarchical methods produce a hierarchy of clusters, where each cluster is nested into another. The hierarchy, usually presented in a dendogram, can be considered as bottom-up or top-down. Both techniques are met in literature and known as agglomerative and divisive methods respectively.

Agglomerative methods begin with n clusters for a dataset of n objects. Then, they choose and merge the closest two clusters into one. This process is repeated until only one cluster is finally remained. On the other hand, divisive methods begin clustering using the opposite direction. They first put all n objects into one cluster and then they split the initial cluster into two smaller. In each step, a cluster is chosen and split up into two. This process continues until n clusters are produced. Those two methods are known for their weakness to revise a previously taken decision. Once a cluster is merged or split, it can never be separated or regrouped. That irrevocable decision is their major defect. Furthermore, the complexity of agglomerative clustering is, in the general case, $O(n^3)$, which makes them inappropriate for large datasets. Also, divisive methods have a worse complexity of $O(2^n)$ in the task of exhaustive search.

In the field of document clustering, many agglomerative methods have been proposed [26, 27]. Among them, the single linkage method SLINK [28] computes the similarity of two clusters as the similarity between the most similar pair of documents each of which belongs in different clusters. The complete linkage method CLINK [29] defines the similarity of a pair of clusters as the similarity between the most dissimilar documents, one in one cluster, and one in the other. The definition of clusters distance is much stricter than for single linkage. In the latter, two clusters may be forced together due to single documents being close to each other, even though the other elements might not be similar. This observation leads to chaining phenomenon where a small amount of large clusters are created. On the other hand, complete linkage provides large number of small, tightly bound clusterings.

A more extensive survey of hierarchical algorithms as well as a comparative analysis between well known hierarchical algorithms can be found in [30, 25].

2.2.3 Density-Based Algorithms

A wide variety of density based algorithms have been proposed in literature [31, 32]. These algorithms rely on a density-based notion of clusters and they inherently carry the notion of noise. This means that density of points within a cluster is considerably higher than outside of the cluster. Respectively, the density inside the area of noise is lower than inside of any cluster.

The key idea of all density based algorithms is that every cluster has a minimum number of points, where the distance among them is below a specified threshold. A big advantage of that technique is that it does not need an a-priori knowledge of the number of clusters and it is able to identify clusters of arbitrary shape and size. The exposure of non linear shapes structures is based on the concept of density reachable and density connected data. The definitions below create clusters of arbitrary shapes, like tubes or spheres.

- **Density Reachable** - A point p is density reachable from a point q, if the distance between the two points is below a threshold ϵ and there are sufficient number of points $MinPts$ in the neighbourhood of point q.
- **Density Connected** - A point p and q are said to be density connected if there exist a point r which has sufficient number of points within its neighbourhood and the distance between the points of p and q is less than ϵ.

Furthermore, density based algorithms are built under the assumption that clusters of low density can be considered as outliers. This assumption can lead into high tolerance of noise. On the other hand, the quality of the clustering is directly related to the density among the points. Therefore, high density varying data is difficult to be clustered and are subject to misclassification.

The most popular method of density-based algorithms is DBSCAN (Density-Based Spatial Clustering and Application with Noise) [31]. The main steps of the algorithm are:

1. Two parameters are required as input, MinPts and ϵ. Then, the algorithm finds an unvisited starting point.
2. It finds the neighbourhood of the starting point within distance ϵ.
3. If the neighbourhood consists of more than MinPts points, then a new cluster is formed. The starting point as well as its neighbourhood are added to the cluster. The starting point is marked as visited.
4. If a new cluster was created, the algorithm recursively visits all the neighbours and repeats step 2.
5. If the points were less than MinPts, they are considered as noise.

DBScan does not require an a-priori knowledge of the number of clusters and the final clusters might be of arbitrary shapes. Furthermore it is almost independent to the ordering of the data and it needs only two parameters as input, that can be given either manually or can be calculated according to some heuristic ways presented in [31]. On the other hand, the quality of the clustering is directly related to the distance metric function. Euclidean distance is the most commonly used metric, but it tends to be insufficient for high dimensional data of sparse

missing point dimensions, as it calculates distances based equally on the presence and absence of data dimensions. This metric can be rendered almost useless due to the so-called "Curse of dimensionality", making it difficult to find an appropriate value for ϵ. Furthermore, DBScan can not cluster data sets with large differences in densities since the minPts-ϵ can not be chosen appropriately for all clusters.

The authors of [32] deal with the problem of varying data density and propose OPTICS, an algorithm of a similar basic idea with DBScan. In order to create meaningful clusters in an environment of varying densities, they linearly order points such that points which are spatially closest become neighbours in the ordering. Additionally, they keep a core distance for each point that describes the distance of the point to its $MinPts^{th}$ point.

Chapter 3

Social Media

In this chapter we provide an overview of social media by discussing several media definitions, we study the special characteristics of the most popular social networks and focus on the informational model of Twitter. We demonstrate a platform built in order to explore and analyze the social content and finally present an interesting statistical analysis deriving from a long period collection of Twitter data.

3.1 Overview

Social media are defined by Kaplan and Haenlein [33] as a group of internet-based applications that build on the ideological and technological foundations of Web 2.0 and that allow the creation and exchange of User Generated Content (UGC). As social media evolve over time new definitions arise, emphasizing in different aspects of the social components. Some definitions highlight the social graph and users communities, while others deal with the generated content or the online tools and applications. However, a common denominator over all definitions is the intention of the users to share content and the impact of this evolution on the research sciences and companies.

Social media exploited the technological evolution and gave the ability to the end users not only to passively consume web content, but also to generate their own. This change dictated a many-to-many communication and led pluralism of opinions to flourish. People are no longer spectators of events but they dynamically participate in an interactive dialogue, they create new events, inform others and express their opinions.

The first form of social media, with the content generated by the end users, have been blogs. A blog is almost equivalent with a personal page, that is usually managed by only one person and contains time-stamped entries of many different variations: personal stories, daily activities, emotional attitudes, political views, technical descriptions are part of a blog's contents. Blogs give the ability to the users to interact with each other by leaving comments about a published post.

Based on the need for more content-based interaction that is not only limited to

the simple textual information, but it also contains multimedia exchange (i.e. photos, video) and instant messaging, many social websites were created. Part of those social sites are built upon the idea of creating a social network among users. They leverage social media to create a two-way communication, building relationships through communities of people with common interests. The main contribution of social networks are the representation scheme of friendship among users (social graph) and the dissemination of information only to the communication channels they define.

Social networks have become very popular in recent years because of the ease and universal access to the internet and the increasing proliferation and affordability of internet devices, such as personal computers and internet tablets. The participation in the social networks is inexpensive, immediate, pervasive and thus massive. Each person is able to distribute and share content and this content is always available and persistent for use. People are able to contribute in an existing on-line conversation, to augment and criticize an article or to delete a comment. With this transparency community can improve and evolve information, propelling its own advancement. Furthermore, the users are independent to post content in real time and in any time, without restrictions or limitations.

In our days we encounter hundreds of social networking sites, focusing on different aspects of social life. Facebook, Twitter, MySpace, LinkedIn, YouTube, Flickr, Delicious and the thematic FoodSpotting, Family Leaf and Next Door are some of the most famous social networks covering a wide range of interests in social life.

Social media introduced a new era in the field of informing. The communication ceased to be one way and people turned from information recipients into producers. As a result, many voices argued that traditional media lost part of their prestige as they assigned to the public their once exclusive right of informing.

On the other hand, many reliability issues were generated as the source and validity of the information can be uncertified. Noisy, untrusted, and uncertain data are problems that users, as well as analysts, are called to address in their activities. The strength of social media to spread UGC in real time can be turned into weakness when unconfirmed or incomplete data are being shared.

3.2 Online Social Networks

It is important to mention that the definition of an online social network is not restricted to an online site, but is more general. In fact, any website or application that allows users to interact with each other and provide a social experience in the terms of exchanging content can be considered as a social network. For example the broad category of social networks contains platforms for photo (Flickr) and movie (Youtube) sharing, movie ratings (Flixster), virtual venues 'check-ins' (Foursquare) through mobile devices and many other applications.

The functional building blocks of social networks can be summarized to seven: identity, conversations, sharing, presence, relationships, reputation and groups.

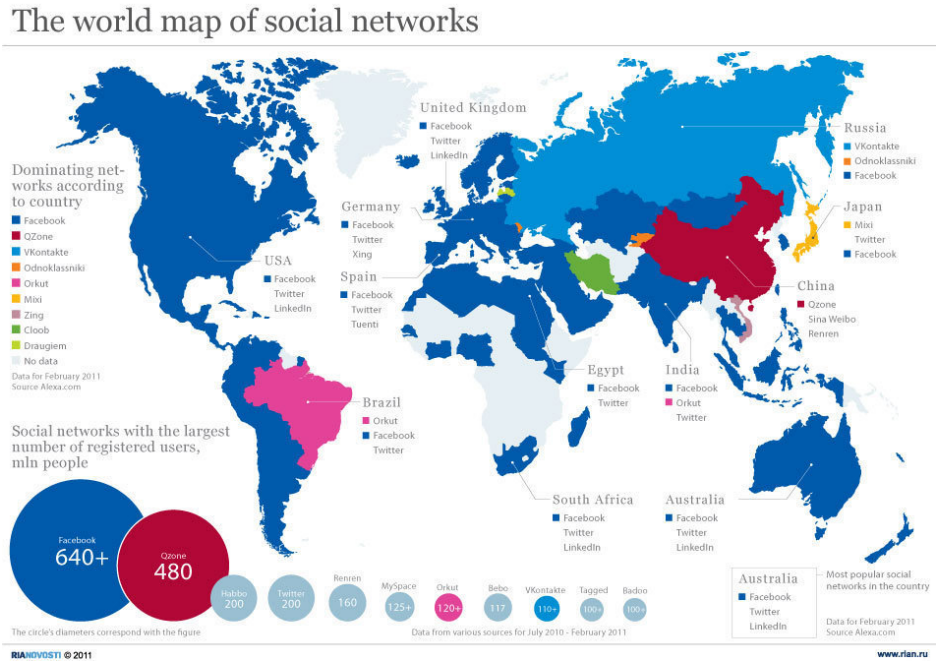


Figure 3.1: Map of social networks' popularity

The identity of users is usually reflected by the creation of an online profile, sharing personal information. Users presence in the social media is succeeded with the status updates or even with social games participation, while conversations are not only limited to comments but also comprehends instant messaging. The functionality of sharing includes photos, videos, articles etc. These building blocks help understand the engagement needs of the social media audience. For instance, LinkedIn users care mostly about identity, reputation and relationships, whereas YouTube's primary building blocks are sharing, conversations, groups and reputation.

Figure 3.2 illustrates the most popular social networking applications by country, according to Alexa [34] and Google Trends for Websites [35] traffic data. Facebook has established its leadership position in 127 out of 136 countries analysed. Although Facebook and Twitter are currently dominating the social networks of the world, there is a social networking site in China known as Qzone which also has the huge size of 480 million registered users. Some other popular networks according to country are VKontakte, Odnoklassniki, Orkut, Mixi, Zing, Cloob, Draugiem. These social networking sites are not so popular in the whole world, but they are very popular in some countries themselves.

The leading place of Facebook and Twitter in the social media market world wide is due to their functionality. As we can see in table 3.1 Facebook provides the biggest functionality compared with the other social applications, as it combines the

existence of a person's profile with the real time communication and social gaming. Twitter sacrifices part of the possible functionality in order to gain simplicity. However, there are many third party applications that build on Twitter's framework and expand it. For instance, TwitPic [36] let users to share media on Twitter in real-time. On the other hand, LinkedIn has a slightly different concept with that of Facebook and Twitter. It is a professional network giving the ability to connect with past and present colleagues and being informed for business opportunities.

Name	Identity	Relations	Presence	Conversations	Sharing	Groups
Facebook	✓	✓	✓	✓	✓	✓
Twitter	✓	✓	✓	✗	✗	✗
LinkedIn	✓	✓	✓	✗	✗	✗

Table 3.1: Basic characteristics of the top 3 social media

In the next section we will focus on Twitter framework. Twitter tends to be attractive not only to the users but also to the analysts and developers, due to its flexible and extensive API [37].

3.3 Twitter Microblogging Framework

Twitter is a popular microblogging platform for social networking, encountering more than 465 million users in the six years of its operation (born July 2006). It offers to its members the ability to connect and communicate with each other sharing short texts, called tweets, of up to 140 characters. The users update their status in real-time, communicating their thoughts, their daily activities or even just watching others people news.

The higher percent of Twitter accounts are public, making the produced tweets available in reverse chronological order to the public time-line of twitter.com's page and exposing a vast amount of data over the internet. Although an account is by the default settings public, users are able to protect their tweets stream by making it private and available only to their subscribed friends.

Few statistics about the growth and innovation of Twitter's service where announced at Chirp developer's conference in April 2010. The first piece of data informs that users accounts are added at the rate of 300,000 a day (2012: 1 million) and they cause an annual growth of 1,500% a year. On a daily basis, 55,000,000 tweets pass through twitter (2012: 175,000,000) with the most of them being originated from United States, Brazil and Japan. Personalities of Twitter that stand out for the big number of followers they have are Lady Gaga, Justin Bieber and Katy Perry. Also, Twitter handles 600,000,000 searches and 3,000,000,000 API requests performed in a daily base.

Although people can interact with Twitter directly through the website, just by logging in to their account, more than 100,000 third party applications have been

created providing extra functionality. As it was revealed 75% of Twitter traffic comes from third-party applications. The ecosystem around Twitter is extensive due to the available Twitter API [37].

3.3.1 Twitter Information Model

The information model of Twitter, slightly outlined so far, is based on two basic entities. Firstly, the users and the relationships among them and secondly the tweets published or republished by them.

The first entity of users consists of name, username, a short description of interests, language and location as well as a unique id for system's internal representation. Further information are carried along with a user's account, like its creation time, the number of published statuses and friends as well as the profile settings (picture, background color etc).

An interesting concept of Twitter platform is the relationship among users. Unlike other social services, such as Facebook or Myspace, the relationships among users need no mutual acceptance or reciprocation. Twitter users can follow others or are followed by others. Being a follower is a unilateral activity, i.e. it means that the user receives the messages from those that the user follows, without necessarily these users being simultaneously informed for his/her recent activity. Thus, relationships between the users are not symmetric, creating a directed graph of users-nodes and associations-edges between them.

The second entity is the textual information exchanged among users. Beyond the 140 characters of each tweet, there is information about its creation time, its source (from web or other application), the id of the user having posted, as well as if it is a reply tweet. A multivalued attribute exists, called contributors, referring to an array of users who authored the tweet on behalf of the official tweet author. Additionally, a new feature is currently activated in Twitter platform concerning geo-location information. The users are able to automatically annotate their tweets with their current geographic coordinates, by enabling the new available option. A set of attributes about the place (name, country name etc) are also available.

Moreover, a common practice on Twitter is the republishing of a tweet, called ReTweet, using the keyword 'RT' in front of the text. A ReTweet expresses user's assent and intention to re-publish the message, causing it to spread faster from one community to another and is a valuable source of information for analysts. Retweets can be considered as an impact indicator, indicating a new upcoming trend or the influence that a tweet might has to other users.

Another interesting concept on Twitter is the use of a set of symbols, facilitating the communication. For instance, addressing the symbol '@' followed by a user identifier in the beginning of a tweet is a reply to the mentioned user, while finding it inside the tweet is just a mention to that user. The usage of '@user' makes the search of related with a user tweets easier and faster. Also, it creates relations among tweets and conversations among users.

Another useful symbol is the hashtag '#'. A hashtag followed by a word repre-

sents a topic and gives the ability to the users to aggregate relevant tweets and to participate to a world-wide conversation. The use of hashtags is a very common practice in Twitter. Some tweets might have more than one hashtags and very often they become trending topics, attracting the attention of the public.

Twitter users also provide links to outside content by including a URL to their tweet. In order to avoid the long length of a URL, people use URL shorteners [1] to generate unique, short and easily shared links.

A graphical representation of the Twitter informational model that captures the concepts, relationships and constraints of the network is shown in Figure 3.2. The basic entities, users and tweets, are depicted as well as the relations among them. We mention that a user can follow and is being followed by many other users denoting a many-to-many relation. Furthermore, a tweet can be re-tweeted many times but it is associated with only one initial tweet and users are able to post many tweets mentioned by only one owner account.

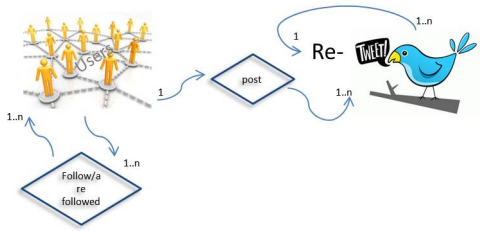


Figure 3.2: Twitter information model


```
"coordinates": null,
"truncated": false,
"created_at": "Thu Oct 14 22:20:15 +0000 2010",
"favorited": false,
"entities": {
"urls": [
],
"hashtags": [
],
"user_mentions": [
    {
    "name": "Matt Harris",
    "id": 777925,
    "id_str": "777925",
    "indices": [
        0,
        14
    ],
    "screen_name": "themattharris"
     }
]
},
"text": "@themattharris hey how are things?",
"annotations": null,
"contributors": [
{
    "id": 819797,
    "id_str": "819797",
    "screen_name": "episod"
}
],
"id": 12738165059,
"id_str": "12738165059",
"retweet_count": 0,
"geo": null,
"retweeted": false,
"in_reply_to_user_id": 777925,
"in_reply_to_user_id_str": "777925",
"in_reply_to_screen_name": "themattharris",
"user": {
"id": 6253282
"id_str": "6253282"
},
"source": "web",
"place": null,
"in_reply_to_status_id": 12738040524
"in_reply_to_status_id_str": "12738040524"
```


Figure 3.3: JSON representation of a tweet

A human readable data message of a status object is presented in Figure 3.3. The format of the message is a JavaScript Object Notation (JSON) which is a text-based open standard providing the involved fields and attributes of the interchange. An extended documentation of every attribute can be found in Twitter API [37].

In the next subsection we outline the basic concepts of our system's architecture built to collect and explore the Twitter stream.

[1] http://www.tiny.cc , http://bit.ly

3.3.2 System Architecture

In order to explore the Twitter stream we designed and subsequently implemented a framework that supports a data storage system and a number of analysis tools. The systems architecture presented in Figure 3.4 describes the conceptual model that defines the structure, behaviour and views of the framework.

The architecture consists of three major layers where each one communicates with its immediately above one. A top-down analysis of the system brings the service provider functions in the highest level. In this layer, the Twitter provides the developers with rich operations in order to easily collect near real-time stream data through streaming API or short historical data using filter criteria.

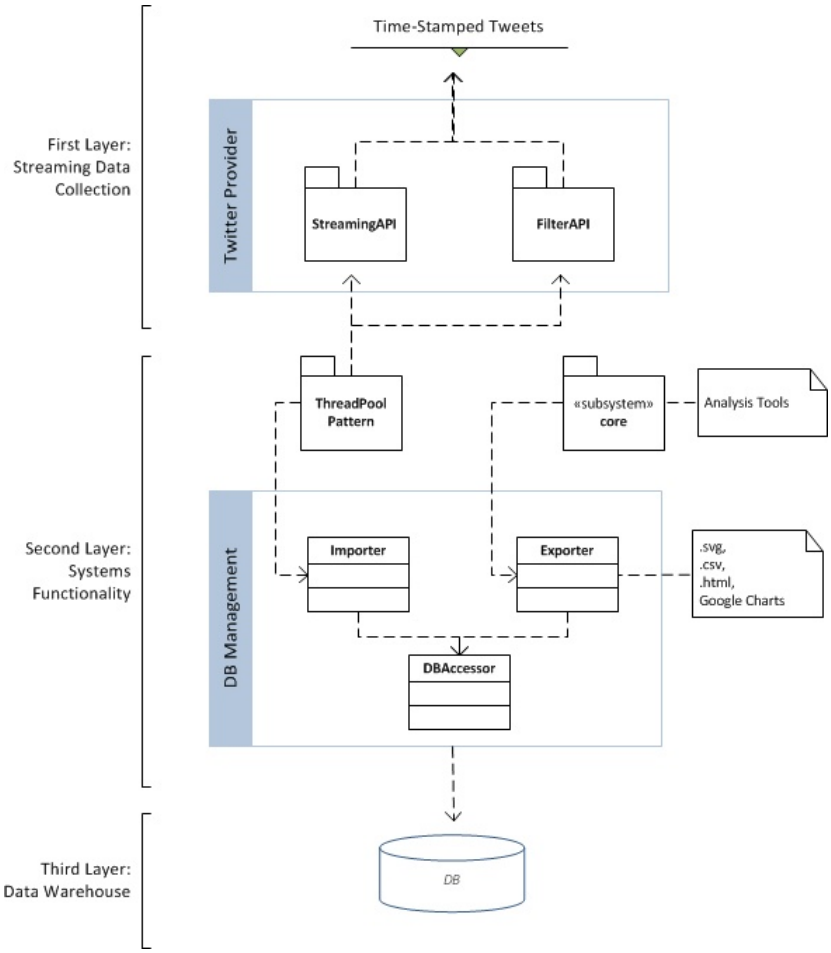


Figure 3.4: System architecture

In the second layer a pool of threads is implemented to manage the incoming data and store them in the data warehouse. The communication with the lowest level is achieved through the wrapper class (DBAccessor) that encapsulates the

functionality of the data warehouse providing a level of abstraction. The Importer and Exporter components give the ability for easily insert and retrieve data in varying formats (i.e. .svg, .csv, .html or Google charts).

The major functionality of the second layer is provided by the core component, which consists of four simpler modules (Figure 3.5). The core package is associated with the Exporter which retrieves data from the database in batches or in incremental mode and feeds them to the operations of the subsequent component. The pre-processing sub-component supports a complete tool for extracting and weighting textual data. The pre-processing procedure includes stemming, stopwords removal and feature selection under different ranking criteria. Furthermore, the well-known weighting schemes TF-IDF and okapi as well as the bursT [21] method are implemented. The third sub-component includes the implementation of the window model techniques (i.e. sliding windows or time-based) for processing data in smaller batches. Finally, the next component includes the implementation of some clustering algorithms, DBScan [31], denStream [38], cluStream [39] and TStream [2]. Last but not least, a variety of supervised and unsupervised quality metrics are available for the evaluation of the algorithms. The set of supervised metrics includes Purity, Precision, Recall, F-Measure and Normalized Mutual Information (NMI) metrics. The set of unsupervised learning supports intra-cluster separation measures, like Between Sum of Squares (BSS) and between cluster cohesion measures, like Within Cluster Sum of Squares Error (WSS) and Silhouette Index.

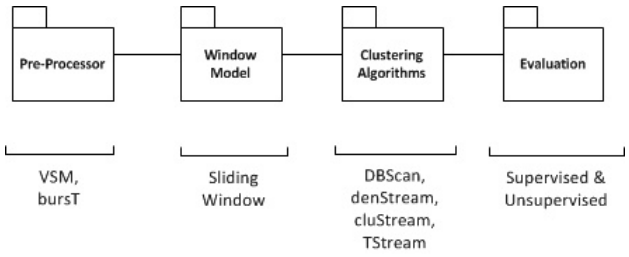


Figure 3.5: Sub-components of the core component

The lower layer contains the data warehouse. The Entity-Relationship diagram is shown in Figure 3.6. A full documentation of each attribute can be found in Twitter API [37].

In the next subsection we introduce the collected Twitter dataset with its basic characteristics, as well as a number of performed measurements part of them already mentioned in the bibliography and verified with our test-bench.

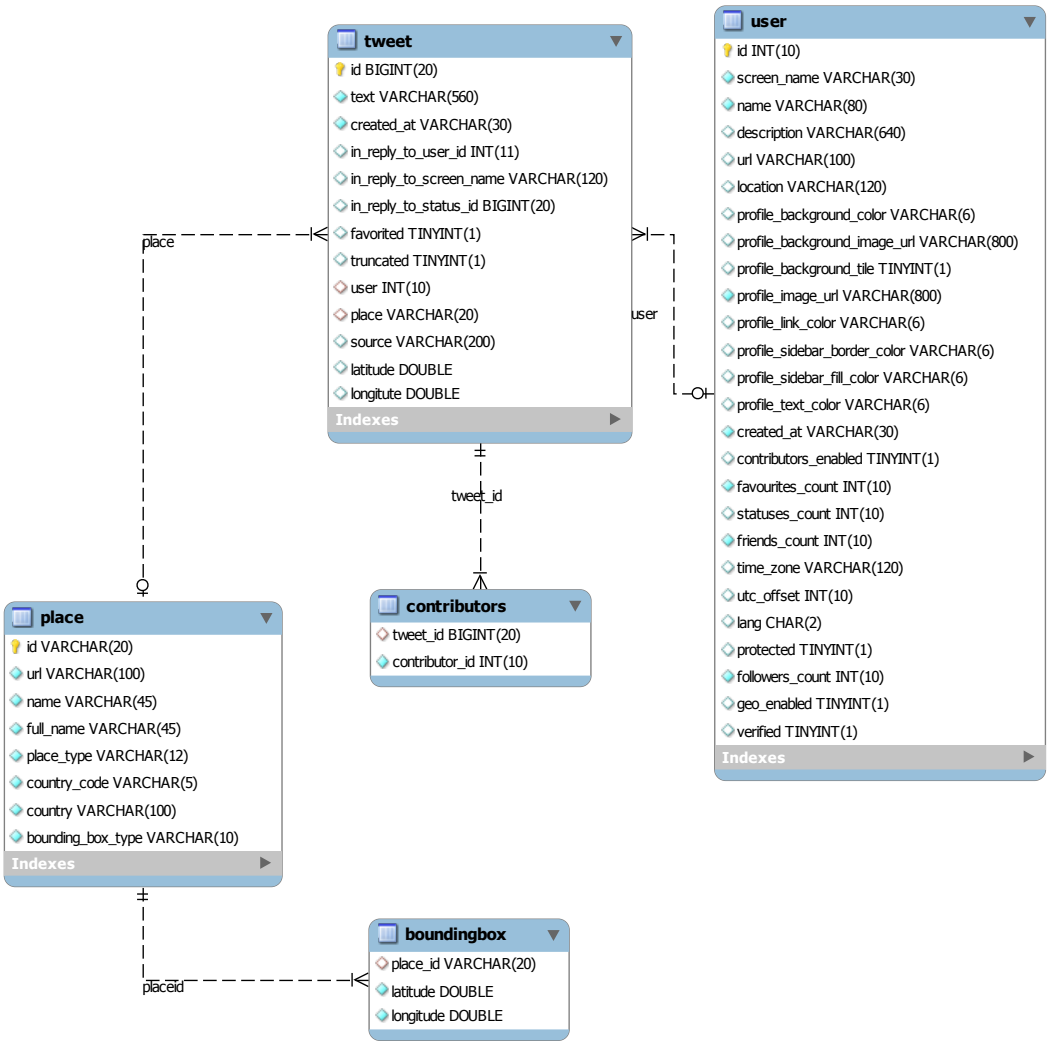


Figure 3.6: An Entity-Relationship diagram of Twitter

3.3.3 Statistical Analysis

Dataset

Our dataset was collected during a 10-months period spanning from September 2nd 2010 until June 9th 2011. The data was collected through Twitter's streaming API [40] using twitter4j [2] java library and is thus a representative sample of the entire stream. Java library is allowed to be served up to 1% of whatever amount of tweets are in the public stream per a "streaming second". The chart 3.7 reveals January as the month of the highest acquisition rate.

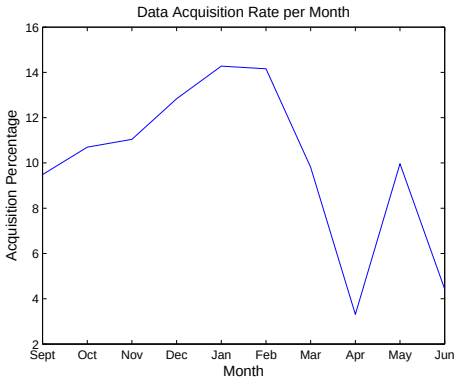


Figure 3.7: Acquisition percentage per month

Table 3.2 summarizes some basic statistics specific to the collected Twitter dataset. From the first two rows of the table we mention that almost 214 million tweets were published from 24 million users, assigning 9 tweets per user on the average. The number of distinct places corresponding to geo-tagged tweets collected through the same period is shown in the third line.

	Distinct Number
Tweets	214,395,334
Users	23,725,151
Places	120,414

Table 3.2: Basic statistics over our Twitter dataset

Furthermore, almost half (47,9%) of the collected users accounts were created during the year 2010, while the lowest percent corresponds to the older accounts (see figure 3.8). This observation shows an upward trend in the creation of new accounts. We avoid commenting on year 2011's percentage as the collection stopped in the middle of the year.

[2] http://twitter4j.org

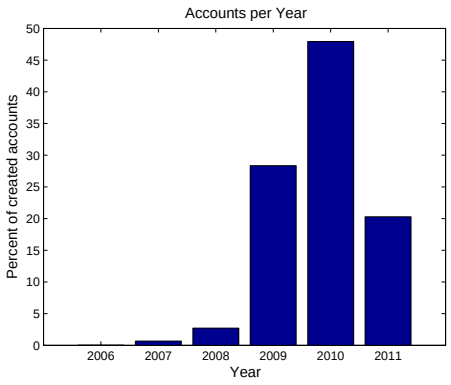
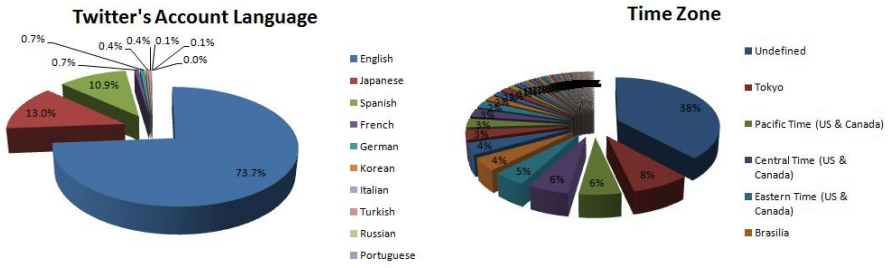


Figure 3.8: Percentage of new accounts per year

To systematize the presentation of our results we present, in the following first section, some statistics that concern Twitter users, their habits, their electronic profile and we analyse, in the second section, the tweets as a meaningful textual information. The measurements derive from the analysis of the collected tweets content and their meta-data. Part of these statistics have been already presented in literature [41, 42, 43, 44] and are verified with our workbench.

Users

Twitter seems to be a multilingual framework attracting and hosting users of different languages (figure 3.9). The largest percent of users, almost 74%, have chosen English as their account language while the 13% and 11% are Japanese and Spanish languages respectively. The rest 2% are divided into French, German, Korean, Italian etc. Also, the largest percentage 8% of users with declared time zone belongs to Tokyo.



(a) Language (b) Time Zone

Figure 3.9: (a) Twitter users language (b) Twitter users time zone

More specifically, most of the users have declared as their current and perma-

nent location the country of Brasil, Indonesia and the town of London. However, the real place of origin of a tweet may change dynamically as users move. This is a challenging problem that Twitter tried to face up with the release of a new geo-location feature on November 2009. Twitterers are now able to automatically annotate their messages with their exact co-ordinates, by enabling the relevant option. However, less than one percent of the users (0,07%) have activated this new and optional feature of Twitter, but we can safely predict that this percentage will raise as third party applications encourage users to use it.

In fact, a remarkable percentage of users use third-party applications for tweeting (figure 3.10). UberSocial is the most popular among them been able to display any embedded link - web pages, blog posts, images and video - alongside the tweet. The second place belongs to the 'Twitter for BlackBerry' application with a 5,51% of twitter popularity, while TweetDeck reaches a percentage of 4,9%. TweetDeck provides applications for desktop, iPhone and Android and from May 2011 is part of Twitter.

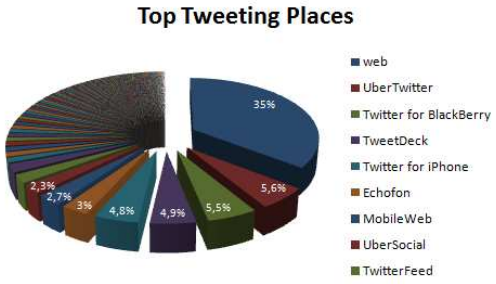


Figure 3.10: Top third-party applications

Coming back to the discussion of the location annotated tweets we see that geo-location information can be used in different levels of granularity. Users can choose between Point of Interest (POI), neighborhood, city, admin or country in order to attach a place in their tweet. The highest percent of users prefer to determine their exact POI, but they hesitate to declare their neighbourhood (see Figure 3.11).

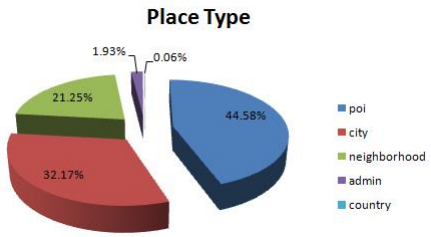
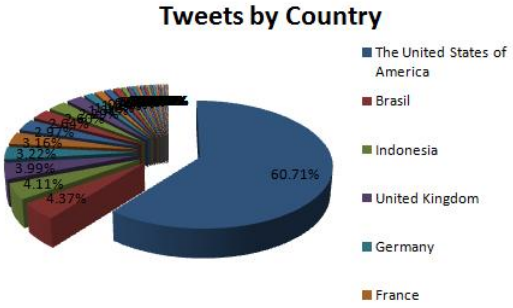


Figure 3.11: Granularity level of place

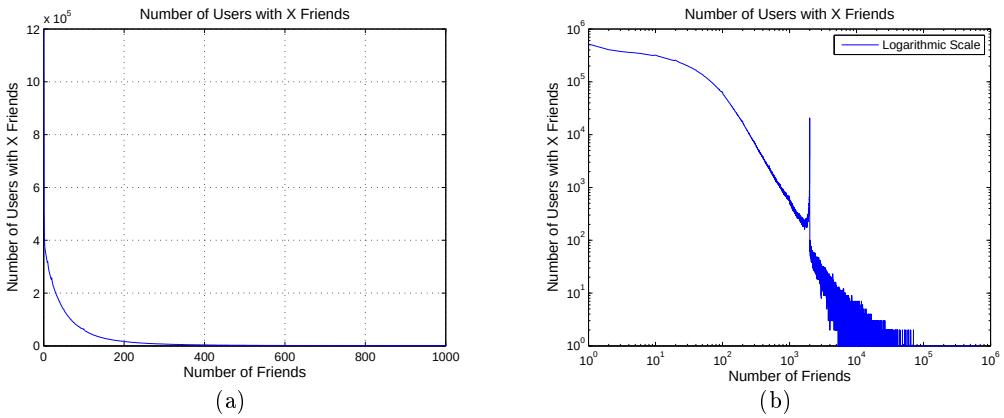
This is not contradictory if we consider that people easily notify that they are at Starbucks, at Subway or at Dunkin' Donuts (top founded POIs), but they do not feel safe to announce their home address. So far, people from United States of America, Brazil, Indonesia and United Kingdom are more familiar with the idea of sharing their position and easily finding others people location. The table 3.3 shows the top twitter countries with the most carrying coordinates tweets, as well as the top cited neighbourhoods.



Neighbourhood Name
Downtown, USA
Highland Park, USA
West End, USA
Riverside, USA
University, USA
College Park, USA
Central Business District, USA

Table 3.3: (a) Top cited countries, (b) Top cited neighbourhoods

Having an in-depth look into Twitter users we need to examine some basic characteristics, such as the distribution of the number of followers (people that follow me) and followings (people that I follow) and the number of published tweets and replies. We mention that only a few number of users have many people that they are following, while the majority has zero followings. Specifically only 1,9% of users have more than 1,000 friends, that they are following. Most of the users (87%) have less than 200 friends.

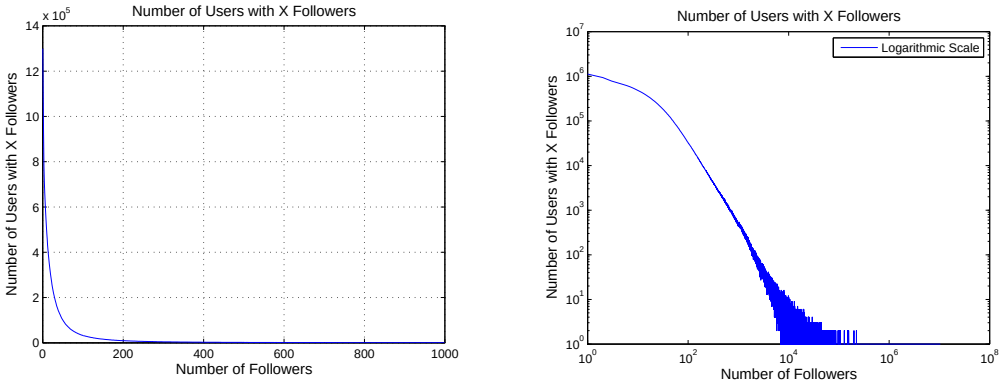


(a) (b)

Figure 3.12: Number of users with X followings

The curve of the figure 3.12(a) reveals that as the number of followings increases the number of users having these followings reduces. We use a logarithmic scale 3.12(b) to reduce the wide range of values to a more manageable size. In general, data have a linear behaviour except the interval $[10^3, 10^4]$ where an intensive peak appears. Particular when the number of followings is equal to 2,001 the number of users increases by an order of magnitude. The explanation of this sudden peak is based on a following limit established from Twitter. This means that a user can not be following more than 2,000 people unless it has the same amount of followers. This barrier exists to avoid a lot of things, among which, is spamming. This limit actually works on a percentage level (10%), but only when a user is almost following 2,000 people.

A smoother behaviour is met in the curve of the distribution of the number of followers (figure 3.13). We mention that only a small percent of users (1,83%) has more than 1,000 followers, while 91% has less than 200 followers. Figure 3.12(b) shows the distribution in a log-log scale format depicting a linear behaviour. The most famous personalities of Twitter with the higher amount of followers are Lady Gaga, Britney Spears, Taylor Swift, Ashton Kutcher and Barack Obama.



(a) (b)

Figure 3.13: Number of users with X followers

In order to gauge the correlation between the number of followers and that of followings, we plot the number of followers (x-axis) against the average number of followings (y-axis) in a log-scale figure 3.14(a). For less than 1,000 followers the average number of followings are almost a reflection to their image in y-axis. For more than 1,000 followers the average number of followings varies in the interval $[1, 10^5]$, indicating that there are users following to others to far more or less than the average.

When looking at the friend (x-axis) and follower (y-axis) relationship 3.14(b), the numbers are fairly balanced because of the Twitter's follower limit rule. As mentioned before, Twitter following limits are based on the ratio of the number of

people you follow to the number of people who follow you.

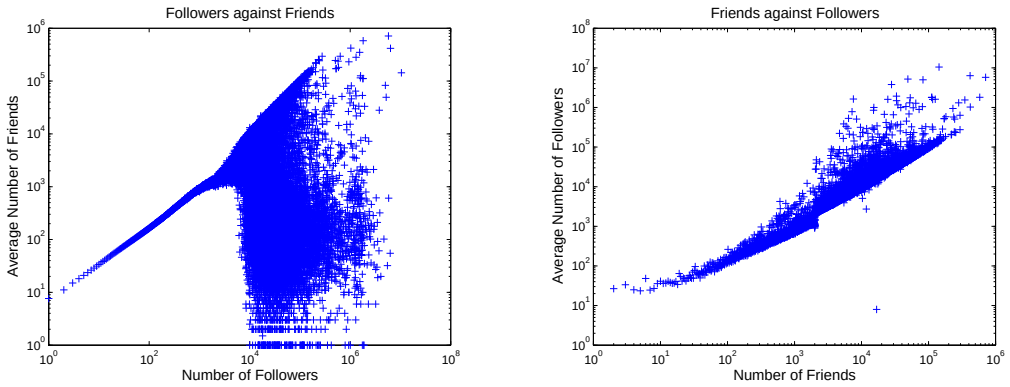
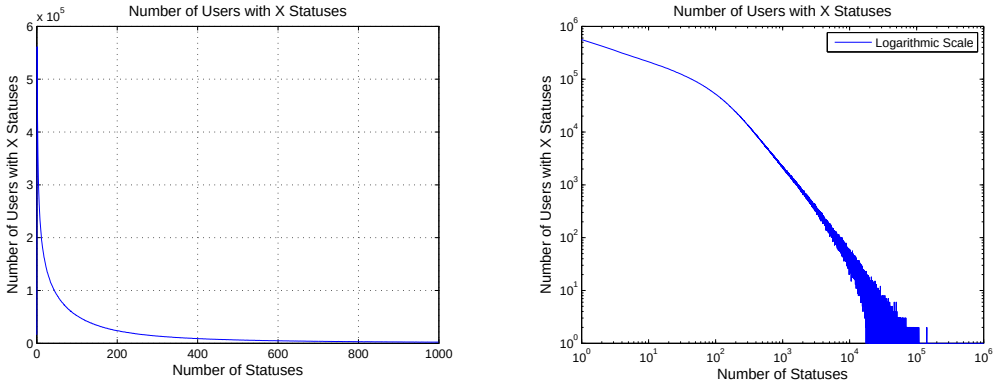


Figure 3.14: Correlation between followers and followings

To conclude, we could summarized users' profiles into three points, based on the proportion of followers and followings. The first category contains all those users that the number of followings overcomes the followers. These users tend to listen others people's opinions and have a small impact with their tweets. The category met in the opposite side contains those users that are famous personalities in the social network of Twitter, or even in the real life, and they share their posts with many followers. This attitude is usually met in famous personalities or news media. The last category is the most balanced, in the terms of the number of followers and followings, while they have a good proportion of 'friends' and a mutual communication. In this category belongs the majority of users that form communities and share posts of varying content.

Another perceptive for distinguishing users' profiles is based on their status updates and their frequency. Most of the users (49,85%) have less than 100 tweets posted, while only a small percent (6,95%) has more than 2,000 status' updates. The curve of the figure 3.15(a) reveals that as the number of tweets reduces the number of users having posted those tweets increases. The distribution is so extreme that if the full range was shown on the axes, the curve would be a perfect L shape. Figure 3.15(b) shows the same plot, but on a log-log scale the same distribution shows itself to be linear. This is the characteristic signature of a power-law distribution.

The five users with the most status updates are dragtotop (search engine), ItIsNow (generator of time), Aviongoo (aircraft market), ThinkingStiff (user) and illstreet (fm radio).



(a) (b)

Figure 3.15: Number of users with X posts

Tweets

Twitter posts are a rich source of information, even though some times exhibit low quality (syntactical and grammatical errors) and contain spam posts (spamming trending topics to grab attention or repeatedly posting duplicate updates). According to their textual composition they can be grouped into four categories, differentiating them from spam and noisy uninformative tweets.

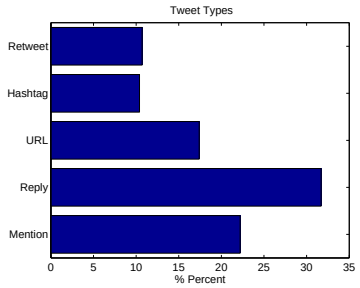
The first category contains tweets that reference to other people (defined by the use of "@" followed by text). The use of "@" associates a tweet with another user, whether answering their message or directing a comment to them. 31,7% of tweets reply to other users, while 22,2% mention at least one other user in their text body. The top five mentioned accounts are @justinbieber, @youtube, @addthis, @soalcinta and @mentionke.

The second category consists of tweets with links to URLs you can visit (defined by the use of "http://" followed by text). A percent of 17,4% of tweets contain at least one url and the percent slightly increases if we include the shorten links (i.e links of the form "bit.ly/" followed by text created by bitly.com site).

Hashtags (defined by the use of "#" followed by text) belong to the third category. They connect tweets with topics enabling the exploration and participation in global conversations. 10,4% of total tweets contain at least one topic and we have detected 4,332,064 distinct topics. Some of the cited topics become, through a Twitter's mechanism, trending topics appearing in the homepage. In our collection the most popular topics are #ff (Follow Friday, suggest a person to the public), #nowplaying (to denote what the user is listening to), #np (now playing), #teamfollowingback (encourage others to follow him/her back) and #fb (Facebook, tweets ending in #fb are automatically imported to Facebook). All of them are being used in every day activities but they are not considered as trending topics.

In the last category we meet re-tweets, passing along information (aka "RT" in the beginning of the tweet followed by text) in a percent of 10,7%.

The table 3.4 shows the relative number and percentage of tweets that fall into each of those categories.



	Number of Tweets
Replies	68,111,427
URL	37,283,498
Topics	22,216,392
Re-Tweets	22,945,024

Table 3.4: Type of tweets

We continue our study by gauging the correlation between the number of posted tweets of a user and its followers (figure 3.16). A similar behaviour with 3.14 is appeared in this chart. The majority of users who have less than 10 followers have tweeted less than 100 times. For users with more than 1,000 followers the number of posted tweets varies with outliers tweeting far more than expected from the number of followers. The chart depicts a correlation, but not a causal relation among the number of followers and their average number of tweets.

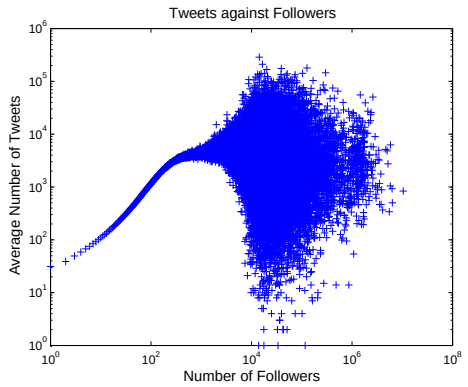


Figure 3.16: Number of tweets against the number of followers

To conclude we have exploited a collection of almost 214 million tweets and 23 million user accounts in order to analyse users profiles and their posted content. We have revealed some interesting habits of the users and some statistics concerning the textual information. We discussed the correlation of followers and followings and how the number of followers affects the posted tweets. Furthermore, we found a power-law distribution of the number of users with x statuses.

Some work exploring the content of Twitter can be also found in the literature. An extensive study on users relationships was done in [44], where the authors studied the topological characteristics of Twitter and its power as a social media of information sharing. The characteristics of social activity and patterns of communication on Twitter have been studied in [45]. The paper [42] emphasizes at the importance of the symbol @ as a degree of conversationality. The importance of retweet, as a conversational practice, is also studied in [43]. In the field of sentiment analysis, the authors of the paper [41] analysed microblog postings containing comments about brands and revealed negative or positive sentiments and opinions about the products and companies.

Chapter 4

Related Work on Clustering Streams

In this chapter a review of the most related works with our statistical approach of clustering social content is presented. The review includes two axes. First, we study several on-line streaming clustering algorithms for large scale data, then we review the challenges of working with social content and present some first steps on social clustering algorithms.

4.1 Stream Clustering Algorithms

In recent years, massive amounts of data are generated from various applications such as network monitoring, telecommunication systems, social media etc. Those data arrive in a streaming mode where the stream is formally defined as:

Definition Stream A stream $S : N_o \rightarrow TxQ : i \longmapsto (t_i, p_i)$ is an endless sequence of points $p_i \epsilon Q$ from a d-dimensional input space Q and $t_i \geq 0$ is the arrival time of object o_i.

The observation that data arrive in an on-line streaming mode opened the door for the study of on-line and stream-oriented algorithms. Particularly, in the field of NED the assumption that the entire corpus is available at any time is no longer valid, as the content is generated and arrive dynamically. The transition from discovering news events from a known corpus of documents to the need of discovering events from a stream of data arriving in an on-line manner made the study of a new generation on-line stream-oriented algorithms imperative. We firstly provide some background information about the family of on-line algorithms and then we discuss their connection with streaming algorithms.

- *Sequential processing of input*
  The input data points are fed to the algorithm point by point, in a continuous

way.

- *Temporal processing of input*
  The data points arrive in a temporal manner, where every input point has a relative time-stamp. The oldest points are met earlier in the input than newer.

- *No memory available*
  At the decision time, neither the knowledge from the future is available, nor the memory from the past. The algorithm decides without having the entire input available.

- *The decision is not optimal*
  Without the knowledge of the entire data, the decision of the algorithm may turn out to be non optimal. The quality of decision making is a challenging task.

In an *on-line* mode the clustering decision has to be taken immediately after a new point arrives in a temporal sequence. Furthermore, the lack of memory forces the algorithm to decide without any previous knowledge leading, in most of the cases, to non optimal results. On the contrary, *off-line* algorithms (see Chapter 2) may achieve an optimal solution as the entire data are available during the decision time. Off-line algorithms have a knowledge of the future at each processing step and they are required to end up with a solution. Part of the techniques, used so far to solve the problem of NED, takes advantage of the knowledge of the entire corpus, including them into the category of off-line algorithms. Nevertheless, off-line algorithms assume that unlimited resources are available. This assumption makes them inappropriate for large scale streaming data.

Stream-oriented algorithms exhibit some differences with on-line algorithms, particularly in the way they consider the time parameter. On-line algorithms are interested in the future. They deal with how their processing step affects the future decisions. On the contrary, streaming algorithms decide about their present, by computing a function over a summary of the past elements. Despite their differences some of the characteristics of the on-line algorithms are desirable in the streaming context.

The size of the stream is considered to be unbounded and input items arrive continuously, without any request, as time progresses. The unbounded size of the data poses memory and processing limitations. It forces the algorithms to discard or memorize part of the incoming data, as well as parsing the same element one or few times. The order of the elements are not under the control of the system. Streaming algorithms dealing with on-line streaming data spend limited processing time per input point and maintain a summary of them into memory, as input data exceed the available system memory. An important issue to address when dealing with large-scale data is the number of comparisons of incoming elements that need to implement. A trade off between runtime performance and clustering

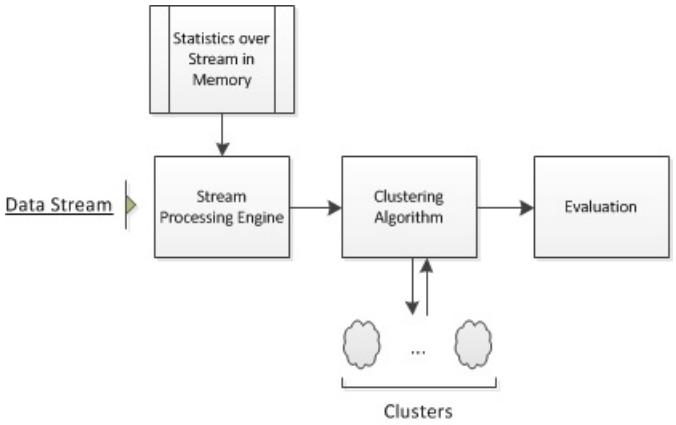


Figure 4.1: Architecture of a stream clustering system

accuracy has to be considered. Two possible solutions have been proposed; the use of statistical properties [46, 47] summarizing the data and blocking methods [48] representing subsets of the data. Both reduce the necessary comparisons over data points. These approaches remind the information extraction techniques (chapter 2.1.2) and share some common characteristics. The algorithms presented below use statistical properties in order to reduce the space of comparisons required by the clustering.

The general architecture of a stream clustering system with statistical representation of its clusters is presented in Figure 4.1. Some statistics are extracted and kept in memory during the arrival of the stream. Several algorithms use time windows [49, 50, 51] in order to reduce the problem of processing the entire stream to the one of processing the most recent items and summarized versions of the old ones. The statistical information is given as input to the clustering algorithm, which produces new clusters or deletes the old ones. The last step of the clustering model is the evaluation of clustering decisions by measuring clustering quality, time and space performance.

BIRCH [46] was the first algorithm published in 1996 for the purpose of clustering massive spatial data. It was the pioneer algorithm for this area awarded by Sigmod conference with the 10 years test of time award. BIRCH algorithm clusters the input data incrementally and dynamically, using a single pass. Each cluster is summarized by a triple, called Clustering Feature (CF). A CF maintains the number of points in the cluster (N), as well as the linear (LS) and square (SS) sum of its points. This CF compact summary is efficient, as it stores much less than all the points and sufficient for calculating all the necessary measurements, like centroid and radius. The proposed algorithm maintains a height-balanced tree, called CF-tree, of the form $[CF_i, child_i]$, where $child_i$ is a pointer to its ith child. A leaf node contains at most L entries of the form CF_i and satisfies the threshold requirement that radius has to be less than T. Such a CF tree is built dynamically

as new data arrive and is used to guide a new insertion into the correct position for sorting purposes.

Many other algorithms were built upon the idea of BIRCH and in some aspects they achieved better results. The first algorithm published after BIRCH was STREAM [52] in 2002. STREAM algorithm processes data streams in batches of points and uses centroids to represent each batch. The clustering process is repeated recursively until it reaches k-clusters. Although STREAM outperforms BIRCH algorithm, they both suffer from the problem of detecting new clusters in an evolving data stream. They treat old and new points as equally important information, ignoring new upcoming trends.

CluStream [39] algorithm was published in 2003 and it also uses CF condensed vectors to store data points. The main idea of CluStream algorithm is to divide the clustering process into two components. The first component, called online, periodically collects and stores detailed summary statistics from the input data in a hierarchical time frame. The second component, the offline, uses the summary statistics and performs k-means clustering, in order to provide a better understanding of the broad clusters. In contrast to STREAM, CluStream reveals trends as it takes snapshots at different time stamps, favouring the most recent data. Also, it maintains a constant number of clusters by removing the less active or by merging the closest ones. The experimental results showed that it outperforms SEARCH algorithm in the terms of effectiveness, efficiency and scalability.

On the other hand, CluStream is somehow limited. First, it needs to know the number of clusters a-priori and it can not find clusters of arbitrary shapes, due to spherical cluster production of k-means. Then, it makes no prediction about outliers and noise and it is time consuming to be applied on large volume data due to multiple passes of k-means.

DenStream [38] proposed in 2006 overcomes weaknesses of CluStream following the structures of CF vectors and the online-offline rationale. It introduces the concept of a fading function which gradually discount the history of past behaviour. Furthermore, it is based on DBScan density-based algorithm and it captures clusters of arbitrary shape. It has the ability to handle outliers and it provides memory and time guarantees.

Prior to DenStream, a worth mentioning algorithm HPStream [53] published in 2004 focuses in another important aspect of streaming data, their dimensionality. The algorithm employs a data projection method to reduce the dimensionality of the data stream to a subset of dimensions that minimize the spread along the chosen dimensions. The experiments showed an improved clustering quality than cluStream in both synthetic and real datasets. Although the dimensionality awareness, the experiments and the dimensionality test were applied in less than 80 dimensions. As we will discuss in the next chapter the number of dimensions is 2-6 orders of magnitude smaller than in the context of social content.

In the footsteps of DenStream lies also the D-Stream [54] algorithm published in 2007. D-Stream exhibits some similarities with DenStream in that it uses also a density based approach, an exponential decay function for ageing and checks

periodically for noise or outdated regions. Also, it is divided into two steps. During the online step it maps each input data record into a grid and during the offline step it computes the grid density and shapes the final clusters.

The following-up C-DenStream published in 2009 [55] belongs to the category of semi-supervised algorithms that make use of labelled and unlabelled data and is based on a variation of DenStream. It is a density-based algorithm that includes domain specific information in the form of constraints, namely Must-Link and Cannot-Link. For each pair of instances, involved in any of these links, they produce a constraint between the clusters to which they belong. The constraint has a weight depending on the arrival time and the number of instances between the two involved clusters. The experimental section showed that a small amount of constraints are enough to improve the performance of the algorithm. Of course the only requirement, which is frequently not satisfiable, is to have available such constraints.

The rDenStream [56] is a three step algorithm, published in 2009, based also on DenStream. The key difference of these two algorithms is that rDenStream keeps a track of discarded clusters from which it learns to improve the accuracy of the clustering. Although it presents better results than DenStream it has worse time and space complexity.

The algorithms presented so far belong to the general category of stream-oriented methods. The problematic of these algorithms is how to process data streams efficiently under limited memory. Under these restrictions they provide techniques for summarizing and incrementally clustering data. Although these characteristics are desirable in the concept of clustering social streams they do not take into consideration the peculiarities of the social textual content. Furthermore, the algorithms have been tested over the KDD Cup '99 Dataset, which includes a wide variety of intrusions simulated in a military network environment and designed to distinguish attack, intrusions, and the rest type of the connections. Thus, the dataset contains few numerical attributes (less than 60) compared with the high number of attributes in a social stream.

In the next section, we discuss the characteristics and the challenging issues arriving from the social stream and study the most recent works on clustering social content.

4.2 Social Stream Mining

We are witnessing an unprecedented growth of interest in social media[1] enabling people to achieve a *near real-time information awareness*. Several online networking sites (e.g. Facebook), micro-bloging applications (e.g. Twitter) and Social news (e.g. Digg) produce on a daily basis vast amounts of user-generated content (i.e. textual posts) related to a wide variety of real-world news (personal, political, commercial etc.). Many websites filter and organize media content, providing users

[1] en.wikipedia.org/wiki/Social media

with recent topics through attractive websites [57, 58, 59, 60, 61]. Although these websites manage to make a message aggregation, tools for text mining and topic detection are recommended in order to supply users with news coming from the citizens of the social media community. The automated analysis of such social text streams has already created scientific and business value.

4.2.1 Challenging Issues

A number of challenging issues arise in mining social content [6]. In an attempt to summarize the peculiarities of social text stream, for example Twitter stream, we could say that Twitter posts are:

1. Short & low quality

   Tweets are, by design, short texts of up to 140 characters with lot of abbreviations and social media slang. Moreover, they often exhibit low quality (syntactical errors, ungrammatical sentences, spelling mistakes) and contain spam posts (spamming trending topics to grab attention or repeatedly posting duplicate updates). Those characteristics have a great impact on the *size* of the extracted vocabulary and its *weighted* representation.

2. Heterogeneous & Noisy

   Twitter users post messages of different type and scale, ranging from personal stories with no interest to a broad audience until breaking news of high popularity. This heterogeneity affects the *number* and the *utility* of recognized clusters.

3. Highly Evolving

   Tweets are characterized by a non-stationary data distribution, as new points arrive over time in a high rate. A cluster's *shape*, *volume* and *density* may be changing over time. This behaviour affects the *memory/performance* requirements of the various clustering algorithms and highlights that the number of clusters and their active period can not be known a-priori.

4.2.2 Clustering Algorithms

Several methods for analyzing social text streams have been proposed during the last years. One seeks to identify ***emerging trends*** [62, 63, 64], that is topic areas for which there is a bursty interest among users. Emerging trends are defined as sets of words or phrases and are typically identified by analysis of the statistics of words co-occurrence.

In [62] the architecture of a system that performs emerging trend detection over the Twitter stream is presented. A real-time and single pass algorithm is used to detect bursty keywords and group them together according to their co-occurrence. Some other parameters, like geographical origins and frequently cited sources, are

taken into consideration in clustering decisions but without providing algorithmic details or experimental results.

An interesting work in the field of emerging detection for a target event is presented in [63]. The authors focus on detecting an earthquake by monitoring a pre-defined set of words, like "earthquake" or "typhoon". Their event detection algorithms use a temporal and a probabilistic model to estimate the event and its location. Their earthquake reporting system detects earthquakes promptly and sends e-mails to registered users. Notification is delivered much faster than the announcements that are broadcast by the Japan Meteorological Agency (JMA).

The field of emerging topic detection is also examined in [64]. The authors detect emerging terms if they frequently occur in a specified time interval and they were relatively rarely occurring in the past. Moreover, they determine users authorities, using the PageRank algorithm, to analyse the social relationships in the network. For every keyword, inside a time interval, a score is calculated as a product of the terms weight and authority value. Then, in order to express the topics related to the retrieved emerging terms a vector of correlations is produced, representing the relationships between the terms. In the last step a term-based topic graph is constructed. Since each topic is defined as a set of semantically related terms, they leverage the topological structure of the topic graph, by searching for strongly connected components, to detect the emerging topics into the Twitter community. Although they introduce the network's structure into the clustering procedure, their method is not applicable to real time and online systems.

A second method seeks to identify and monitor ***topics*** in social streams [65, 2, 66, 67, 68]. In these works, topics are defined as clusters of similar textual posts.

In [65] the authors present a very common practice for the problem of topic detection (see algorithm 1, section 2.2). Based on that technique, they propose a new improved algorithm that applies locality-sensitive hashing on web-scale corpora to solve the topic detection problem over streaming data efficiently. They compare their system against UMass system [69] on the standard TDT5 corpus of documents, achieving an order of magnitude speed up in processing time while retaining comparable performance. Also, they apply their algorithm over a stream of Twitter data but without providing any clustering quality results. Assuming a good quality performance, as proved from the comparison with UMass, they manually annotate clusters into three categories of spam, neutral and event. The gold standard produced from this process was used to evaluate a set of four ranking methods (random, cluster's size, number of users, entropy). They proved that taking entropy metric into account can reduce the amount of spam in the output.

The authors of [2] propose a two-level hierarchical algorithm (named TStream) to detect, track and update large and small bursts of news in a two-level topic hierarchy of broad topics and more specific subtopics. First, the algorithm computes the document novelty by evaluating the cosine similarity between the incoming document and the clusters of the first hierarchy. If the similarity is greater than a specified threshold a merging is performed. Then it further checks if the document can be absorbed by one of its subtopics. If the similarity is again greater than the

threshold, the document is added to the subtopic. Otherwise, a new (sub)cluster is created containing the document. Note that the document novelty is computed with respect to the feature space of the current level. Also, some re-organizations are performed in each level in order to avoid unlimited number of clusters. The evaluation of the algorithm is done over a stream of news articles using different hierarchy updates strategies, but there is no comparison with the existing state-of-the-art algorithms.

An attempt to use the content and community structure in order to create the clusters is done in [66]. The clustering algorithm computes the similarity, as a linear combination of the structural and content-based values, between the document and the summary of the cluster. Also, a supervised version of the algorithm is proposed where historical data are kept. The experimental evaluation is performed over two datasets, Twitter and Enron email stream. They showed that the use of content- and network-stream based clustering has a number of advantages in comparison to pure text based methods.

A different perspective represents the social text stream as a graph [67], where social actors are the nodes and the information flow between the actors represents the edges. Each corresponding edge embeds the content and the temporal associations between the flow of information. Using the above multi-graphs and text based clustering they detect events, producing better results than the state-of-the-art event detection algorithms.

A similar graph-centric approach is presented in [68]. The authors suggest a graph-based representation of the textual content, where the nodes indicate selected keywords of high document frequency and edges denote high co-occurrence probability between the connected keywords. Assuming that the keywords co-occur when there is a meaningful topical relationship between them, the authors calculate the betweenness centrality of each edge and extract the underlying communities. A slight variation is applied to the community detection algorithm in order to support multiple occurrences of the same node in different communities. They have tested their system with different definitions of keywords, such as nouns or named entities but they present no quality metrics for clustering. Furthermore, no discussion is made about the complexity of the algorithm and its scalability to real time event detection over text streams.

Other related works focus on identifying ***real-world events*** along with their date and time, participants, and location [70, 71, 72].

Real world events are detected in [70] exploiting the tags supplied by Flickr users to annotate photos. In particular, the temporal and locational distributions of tag usage are analyzed in the first place, where a wavelet transform is employed to suppress noise. Then, a detection of aperiodic and periodic events is performed and event-detected tags are clustered into clusters of different events.

In [71] some effort is made to explore the rich content associated with social media and a variety of similarity metrics based on textual, time, date and location features are proposed. Distinguished clusters based on these similarity functions were created and a weighted ensemble clustering approach was proposed to combine

the results. The experiments, over a dataset of Flickr photographs, suggested that the learning techniques yielded better performance and gained a significant improvement over traditional content-based similarities.

An exploration of Twitter content is performed in [72] where a distinction of events and non-events is discussed. An event is associated with a real-world event of a time period and a number of tweets, while non-events are Twitter-specific conversations, memes or retweet activities. The authors propose an on-line clustering and filtering framework where a threshold-based incremental algorithm is applied using vector space model and similarity function. There is no discussion concerning the parametrization of the algorithm and how a static weighting scheme can be scaled to real time clustering. The main objective of the paper is the distinction of event and non event clusters. The decision that a cluster corresponds to an event is taken through a classifier trained with temporal, social, topical and Twitter-centric features. The proposed classifier outperforms in quality two other simpler classifiers where the choice of event-cluster is randomly or based on the fastest-growing cluster. The evaluation of their system is based on the precision without considering the impact of existed events that have been missed.

Other aspects of social stream analysis have also been studied, such as social recommendation [73], community detection [74] and information diffusion [75] but is beyond the scope of out work.

The works presented in this section deal with the social content either by analysing the statistics of words co-occurrence or by clustering textual data in batches, partially exploiting the knowledge of stream-oriented algorithms (section 4.1). Although the clustering procedure is enriched with the social content (time, geo-location, tags etc) it exhibits some limitations. It does not capture the evolution of the stream and it can not maintain clusters of evolving shape and centroid. This limitation causes scalability issues on real time clustering applications. In the next chapter we study the parameters and the properties of the clusters to get some valuable insight of the clustering procedure.

Chapter 5

A Realistic Workbench

Although most of the current works are experimentally validated there is still no systematic workbench that takes into account the peculiarities of social streams exhibited in reality enabling us to benchmark different kinds of analysis algorithms in an unbiased way.

In an effort to achieve a better understanding and insight of the social stream data and their effect on the analysis tools we designed and subsequently used three data-samples sampled from an initial corpus of 9,062,914 English tweets, collected under Twitter Filter API [37] using a variety of tags. Out of this collection we have extracted three sampled datasets, using the Alias method [76], exhibiting different evolving behaviours (in terms of arrival rate and volume) as well as topic heterogeneity.

In the next sections we discuss the characteristics of the initial corpus that consists of two datasets and present the sampling methodology for the creation of the three representative samples. Furthermore, we discuss the peculiarities of the stream captured by each sample.

5.1 Datasets

The initial corpus consists of two datasets collected under different filter criteria and much varying arrival rate and volume. The English tweets were manually extracted from both datasets utilizing an English dictionary and maintaining only the tweets with more than 60% of English words (after the removal of URLs, digits, usernames and punctuation).

As depicted in Table 5.1 the first dataset consists of four different topics occurring sequentially during a nine months period. Topics have different arrival rates varying from few hundreds to some thousands tweets per day. The arrival rate of each topic is an indicator of its popularity and users interest during the period of collection. Furthermore, the varying filter tags as well as the data volume in percent are shown in Table.

Topic	Tags	Data Volume	Date	Arrival Rate (tweets/day)
Flotilla	flotilla, Gaza	48,939 (1.1%)	10 Sep 2010 - 23 Feb 2011	326
Libya	Gadafi, Tripoli, Libya, Libia	4,246,403 (96.3%)	23 Feb 2011 - 27 May 2011	47,182
Champions League	championsleague, champions league	110,862 (2.5%)	27 May 2011 - 29 May 2011	55,431
European Revolution	europeanrevolution, yes we camp	3,488 (0.1%)	29 May 2011 - 09 Jun 2011	317

Table 5.1: Basic characteristics of dataset I

The second dataset, depicted in Table 5.2, involves the topics of the first dataset plus one more topic occurring simultaneously over time. The extra topic is collected during the earthquake and tsunami in Japan under common used tags in Twitter network and appears the highest arrival rate of all the topics.

Topic	Tags	Data Volume	Date	Arrival Rate (tweets/day)
Dataset I	*dataset I*	4,409,692 (48.7%)	10 Sep 2010 - 09 Jun 2011	
Japan	prayforjapan, tsunami, earthquake, Japan	4,653,222 (51.3 %)	11 Mar 2011 - 24 Mar 2011	332,373

Table 5.2: Basic characteristics of dataset II

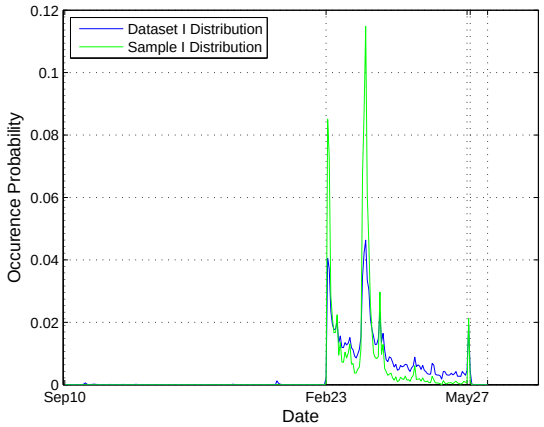
5.2 Sampling Methodology

Dealing with large amounts of data leads to the need for a smaller representative sample where the measurements and experiments require less time. In this section we discuss a sampling technique for creating samples that take into consideration the particular characteristics of social content and benchmark different aspects of the clustering algorithms.

1. Sample I: Ageing Behaviour

   The first sampling method retrieves tweets from the whole body of the dataset I maintaining the tweets' occurrence distribution over time. For this purpose, the alias [76] method was used. The alias method is a family of efficient algorithms for sampling from a discrete probability distribution. The algorithms typically use $O(n \log n)$ or $O(n)$ preprocessing time, after which random values can be drawn from the distribution in $O(1)$ time. The distribution of the sample, as well as its special characteristics (distinct words and average words per tweet) are shown in Table 5.3.

   The first sample was created to study the effect of different arrival rates and volumes between topics of different time periods. The sample was created with respect to the distribution shape over time of dataset I.



Sample Statistics	Topic	Size
Distinct Words: 16,029	Libya	9,771
Average Words /Tweet: 11,4	Champions League	229

Table 5.3: Distribution shape over time & characteristics of sample I

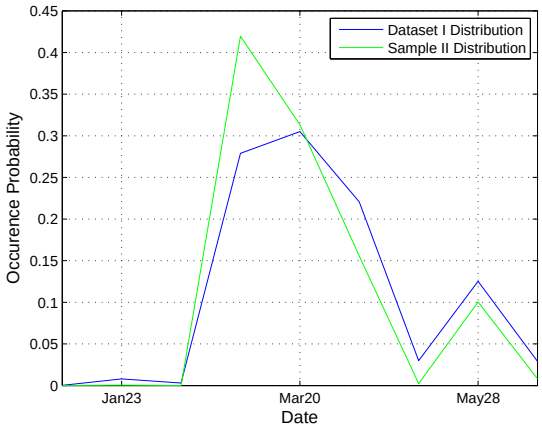
The goal of this sample is to force clustering algorithms to capture the different topics when varying arrival rates exist. The dataset simulates an ideal situation where there is no chronological overlap among the different topics, as they arrive in distinct time intervals. However, the amount of data and the duration are much different per topic. The ageing behaviour of each clustering algorithm, as well as their ability to remember as much information as is needed are two parameters examined by this sample.

2. Sample II: Bursting Behaviour

   An alternative method employed in order to extract sample II. Tweets were collected from around the burst period of each topic. The sampling was done in the time interval from the day before to the day after the day with the highest arrival rate.

   For the purpose of creating this sample,the alias method was used on the probability distribution of dataset II. The distribution shapes of the dataset and the sample are shown in Table 5.4.

   The second sample highlights the effects of the burst behaviour including only the days with the highest percentage of tweets arrival.



Sample Statistics	Topic	Size
Distinct Words: 13,225	Flotilla	5
Average Words /Tweet: 10,9	Libya	8,887
	Champions League	1,108

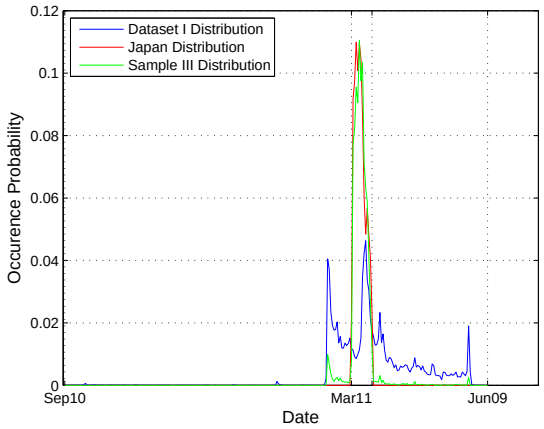
Table 5.4: Distribution shape over time & characteristics of sample II

   The major goal of sample II is to test the ability of the algorithms to detect a topic based on its arrival burstiness. This sample captures the ability of the algorithms to detect that a burst of the occurrence frequency of some terms is enough to create a new cluster. In this concept, there is a danger of overseeing terms that are not yet frequent enough to become a topic of their own, even if the terms forming the feature space are adjustable. Many algorithms in the literature (i.e. denStream [38]) are parametrisable on the tolerance of creating new clusters (i.e. parameter ϵ of denStream algorithm).

3. Sample III: Topics in parallel

   Sample III focuses on the effects which chaperon the topics that overlap over time. The third sample was created with respect to the distribution shape over time of dataset II. The Figure of Table 5.5 illustrates the distribution shape of the sample and dataset II by separately depicting dataset I and Japan topic collection.

   Sample III gives the potential to the clustering algorithms to test their ability to distinguish topics that occur simultaneously. Clustering algorithms need



Sample Statistics	Topic	Size
Distinct Words: 18,828	Libya	2,605
Average Words /Tweet: 10,85	Champions League	26
	Japan	7,369

Table 5.5: Distribution shape over time & characteristics of sample III

to associate incoming data into different clusters based on similarity metrics. The appropriateness of the similarity function is a key issue for all the clustering algorithms.

Chapter 6

Statistical Analysis of Social Text

Most of the algorithms used in the related works are based on clustering techniques, thus we are dealing with a clustering problem forcing us to examine the critical parameters of forming new clusters as well as the properties of the clusters themselves. Then, based on the extracted knowledge we evaluate clustering algorithms over the three samples.

In this Chapter, we introduce the critical parameters of the clustering process. Particularly, we compare different *representation schemes* (i.e. $tf \cdot idf$, *okapi* etc), *distance metrics* (i.e euclidean, cosine similarity etc) and *number of dimensions*.

Since the parameters are experimentally estimated, we study the properties of the clusters emphasizing on the investigation of *cluster's centroid*, *shape* and *density evolution* over time. The centroid of a cluster summarizes the discussion of a topic by providing the vocabulary consisting of the most representative words. The various opinions of the users as well as their opinion convergence or discrepancy over time are illustrated in the cluster's shape and density. Our intuition is that the textual social content of dynamic nature results in an evolving vocabulary and opinions of varying flavours. Therefore, the clusters are shifting in space over time with shape shrinking and expanding dynamically. Thus, we assume that the centroid and shape of the clusters are not constant but evolve over time. We test our assumption by applying statistical hypothesis tests and discuss the experimental results. Furthermore, we explore the cluster's density over time by providing a way to detect shrinkage or expansion of the clusters shape.

The understanding of cluster's behaviour in time and space is a challenging task when dealing with social content and can become a powerful tool for clustering algorithms. All of the prototypical clustering algorithms represent, with slightly different patterns, the centroid of the cluster and impose a shape in data. An a-priori knowledge of the centroid behaviour and cluster shaping in time and space can contribute to the decision of selecting the best clustering algorithm for the given problem. To the best of our knowledge this is the first work that tackles this issue.

6.1 Clustering Parameters

In order to examine the different clustering parameters we calculate Pearson's correlation coefficients between the proximity matrix and the ground truth. The proximity matrix summarizes the distances between each pair of tweets as calculated from a given distance metric function, while the ground truth matrix represents the real distances with zero value for tweets of different clusters-topics and one for tweets of the same cluster.

Pearson's correlation, represented with ρ, is a measure of linear dependence between two variables giving a value between -1 and +1 inclusive. It is defined as the covariance of the two variables divided by the product of their standard deviations.

$$\rho_{X,Y} = corr(X,Y) = \frac{cov(X,Y)}{\sigma_X \sigma_Y} = \frac{E[(X-\mu_X)(Y-\mu_Y)]}{\sigma_X \sigma_Y}$$

The correlation quantifies the cluster tendency of the data. It measures whether tweets in the same topic tend to have smaller distances with each other than with tweets in other topics. Higher values of correlation indicate bigger similarity between the sample and the ground-truth and thus a good set of parameters for cluster distinction.

The Figure 6.1 is a representation of the two matrices (i.e proximity and ground-truth) and serve as the direct perception of the relationship between them. They are projected through suitable color spectra to construct corresponding matrix maps in which each matrix entry (i.e distance metric) is represented by a color dot. The left panel of Figure 6.1 shows the proximity matrix of Sample III coded by a red-green-blue spectrum. The right panel depicts the expected and optimal map based on the knowledge of the ground truth.

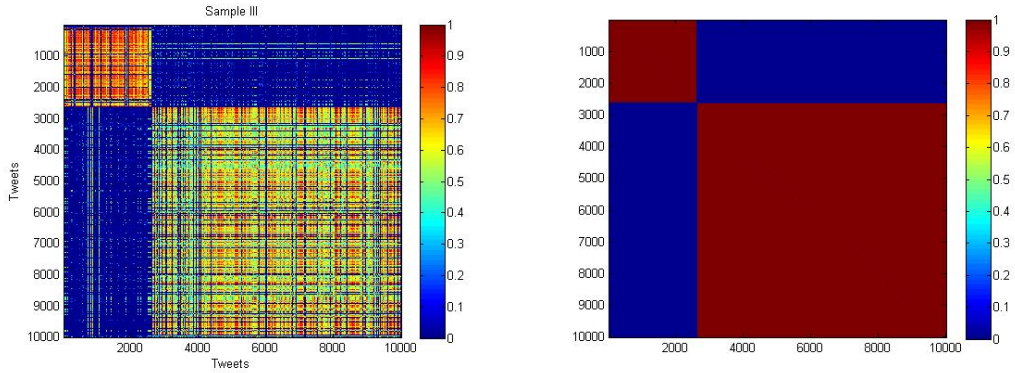


Figure 6.1: Proximity and ground truth matrices

In the matrix map, a red (blue) dot in the ij-th position of the map represents a relative small (large) distance between the tweet i and j. Warmer intensities of color

stand for stronger similarity of points while cooler colors represent no similarity.

The between matrices relations is calculated with Pearson's correlations and scores 66% of correlation between the sample III and the ground-truth matrix, using $tf \cdot idf_2$ weighting scheme (see below), cosine similarity distance metric and 1,000 dimensions.

The comparative study includes two well-known weighting schemes $tf \cdot idf$ (6.1) and *okapi* (6.2). Furthermore, we propose and evaluate a variation of these schemes named $tf \cdot idf_2$ (6.3) and $okapi_2$ (6.4) respectively. The major difference of these schemes with the initial is that $tf_{wi,C}$ refers to the frequency occurrence of the word i (w_i) in the whole corpus C instead of the frequency in each tweet.

$$tf \cdot idf(w_i, t_j) = tf_{w_i,t_j} * log(\frac{N}{df_{w_i}}) \tag{6.1}$$

$$okapi(w_i, t_j) = \frac{tf_{w_i,t_j}}{0.5 + 1.5 * \frac{l_{t_j}}{l_{avg} + tf_{w_i,t_j}}} * log\frac{N - df_{w_i} + 0.5}{tf_{w_i,t_j} + 0.5} \tag{6.2}$$

$$tf \cdot idf_2(w_i, t_j) = tf_{w_i,C} * log(\frac{N}{df_{w_i}}) \tag{6.3}$$

$$okapi_2(w_i, t_j) = \frac{tf_{w_i,C}}{0.5 + 1.5 * \frac{l_{t_j}}{l_{avg} + tf_{w_i,C}}} * log\frac{N - df_{w_i} + 0.5}{tf_{w_i,C} + 0.5} \tag{6.4}$$

The rest symbols are summarized in the table 6.1.

Symbol	Definition
w_i	Word i
t_j	Tweet j
tf_{w_i,t_j}	Frequency occurrence of word w_i in tweet j, t_j
$tf_{w_i,C}$	Frequency occurrence of word w_i in corpus C
df_{w_i}	Number of tweets containing the word w_i
l_{t_j}	Number of tweets containing the word w_i
l_{avg}	Average length of tweets
N	Number of tweets

Table 6.1: Table of symbols

Furthermore, the comparative study includes four well known distance metrics functions used in measuring points distance. Given an m-by-n data matrix X, which is treated as m (1-by-n) row vectors $x_1, x_2, ..., x_m$ the various distances between the vector x_s and x_t are defined as follows:

$$d_{euclidean} = \sqrt{(x_s - x_t) * (x_s - x_t)'} \tag{6.5}$$

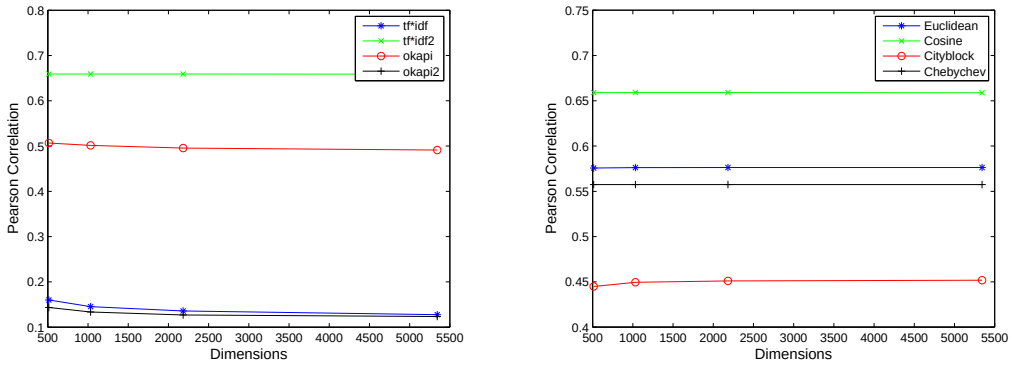
$$d_{cosine} = 1 - \frac{x_s * x_t'}{\sqrt{(x_s x_s') * (x_t x_t')}} \tag{6.6}$$

$$d_{cityblock} = \Sigma_{j=1}^{n} |x_{sj} - x_{tj}| \tag{6.7}$$

$$d_{chebychev} = max_j |x_{sj} - x_{tj}| \tag{6.8}$$

For the comparison of the various weighting schemes and distance metrics the top dimensions of the corpus are extracted. The dimensions correspond to different terms in the corpus selected by two different ranking criteria. The first refers to document frequency criterion, where the terms that occur in the most tweets of the corpus compose the current vocabulary. Thus, each time we increase the size of the vocabulary the new vocabulary contains the terms of the previous one plus the next top terms in the ranking queue. The second criterion ranks the terms based on the weights of the weighting scheme and will be discussed in details later.

As shown in the chart 6.2(a) $tf \cdot idf_2$ using cosine similarity metric exhibits better results than the rest, no matter the number of dimensions. The chart 6.2(b) indicates that cosine similarity metric of $tf \cdot idf_2$ has better results than the other distance metrics.



(a) Cosine Similarity

(b) $tf \cdot idf_2$

Figure 6.2: Comparison of weighting schemes & distance metrics, sample III

The cosine similarity metric is usually used for high dimensional data, where data points are sparse vectors. While Euclidean distance is useful in low dimensions, it does not work as well in high dimensions. Euclidean metric calculates

the distance between two data points based on all the terms even on the missing ones, considering missing and present terms of equal importance. However, in high dimensions, the presence is more important than the absence of a term, provided that most of the data points are sparse vectors (not full). That limitation is being overcome by the cosine similarity where the dot product and thus the angle among the existing attributes specify their proximity.

Furthermore we note that increasing the dimensions over 500 has no significant benefit, while less than 500 results to many zero weighting vectors.

A different ranking criterion of the terms is used to compare the weighting schemes. The terms are ranked in descending mode based on their weights given by the weighting schemes. Only the first two schemes, $tf \cdot idf$ and $okapi$, can participate in the comparison as they assign a global weight to each term. On the other hand, metrics $tf \cdot idf_2$ and $okapi_2$ may assign different weights for the same term according to the frequency occurrence of the term in the given tweet. Figure 6.3 depicts that $tf \cdot idf_2$ has a better discriminative behaviour than $okapi_2$.

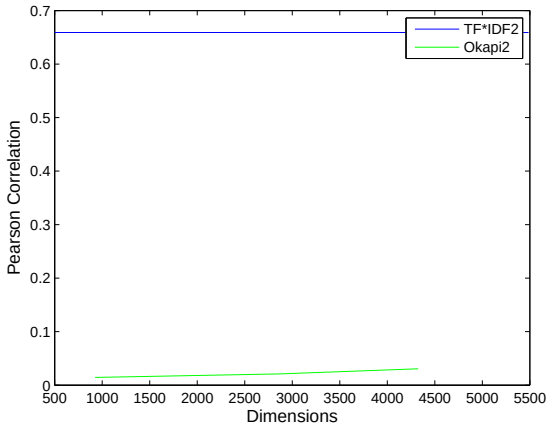


Figure 6.3: Comparison of weighting schemes, sample III

In the rest of our study we perform all of the experiments based on the best clustering parameters found. We select $tf \cdot idf_2$ for the weighting representation, cosine similarity for distance calculations and the 500 most frequent words in the corpus with the highest document frequency values.

6.2 Cluster's Properties

In this section we study the properties of the clusters having already estimated the best clustering parameters. The fundamental properties of a cluster are the cluster's centroid and its shape. We prove for both of them, utilizing statistical learning methods, that they evolve over time with the centroid moving in space and shape shrinking around it. We exploit two different sliding window techniques highlighting their advantages in order to prove our argument and discuss the importance of that evolution for the clustering algorithms. In the first subsection we focus on the centroid trajectory, while in the next we discuss the evolution of the shape and its shrinkage.

6.2.1 Centroid Trajectory

The centroid is a representative point that summarizes the contents of the cluster and defines a geometric center of the cluster's shape. It is not necessarily member of the dataset and the most common technique to define its value is by calculating the mean of the cluster's data points. It can be easily declared in a static and a-priori known dataset, but has to be incrementally updated in an online clustering procedure. The usage of centroid contributes to the efficiency of the clustering procedure as it reduces the comparisons of the incoming object with all the cluster's points to one, that of centroid.

In the context of our problem, the centroid includes the mean of the weights of the most representative words (with the highest df_{w_i}). Therefore, the study of centroid trajectory semantics reveals the evolution of the representative points and thus the evolution of the topic's vocabulary. A non significant movement of the centroid indicates a static topic summary of an already shaped public opinion where the most representative words are repeated in the same volume of posts and are of the same weighting importance. On the other hand, a shift of centroid shows a differentiation in discussion over time with some words obtaining and other missing their significance. Thus, the vocabulary of the users changes as the time progresses forcing the centroid to move in its multi-dimensional space. Centroid evolution is an interesting property of the cluster and a critical parameter of devising proper algorithm for the given clustering problem.

Our hypothesis states that by updating the centroid with the new incoming points its position changes dynamically in space and time in a way that it is unlikely to have occurred by chance. For the purpose of proving our hypothesis and deciding for the statistical significance of centroid evolution we perform statistical hypothesis tests over the three samples. The experiments are done using two variations of the sliding window technique, i.e. counted based and time-based sliding windows. Furthermore, we hypothesize that the evolution occurs even with the use of dynamic weighting schemes. Thus, we apply the statistical hypothesis tests over the bursT [21] dynamic weighting scheme using time-based sliding windows.

Count-based Sliding Window

Assume that a data stream consists of a set of multidimensional points $p_1, ..., p_i, ...$ arriving at time stamps $t_1, ..., t_i, ...$ and $p_i = (p_1^i ... p_d^i)$. In the count-based sliding window model, only the most recent N records are considered at any time, where N is the size of the window. The most recent N records are called active and the rest are called expired records which no longer contribute to the clustering. In the case of our experiments, we introduce a slip step where n_{step} number of new points are added to the window, while the first n_{step} are dropped out. Figure 6.4 shows the contents of the count-based sliding window of a continuous posts stream for two time points (T_i, T_{i+1}).

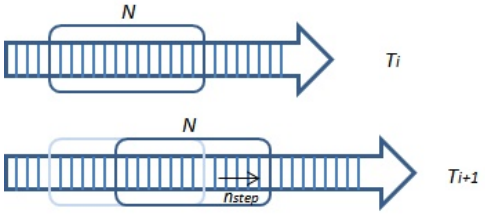


Figure 6.4: A continuous post stream using count-based sliding window

In the first part of our experiments we consider n_{step} equals to N. Thus, we split the samples into partitions $M_1, ..., M_k$ of fixed size windows. For each set of different partitions we hypothesize that the cluster's centroid of M_i and M_j, $i < j <= k$ and $i \neq j$, are equal. Then, we apply a two-sample Hotelling's T-squared test [77] to compare the independence of the two populations.

Specifically, for each sample we select the thematic area of the highest volume of posts (i.e. Libya for sample I) and partition it into segments of maximum 2.000 tweets. Then, we use the first partition for initializing the $tf \cdot idf_2$ representation scheme and the rest to compute the empirical distribution of Hotelling's T-squared statistic.

The two samples Hotelling's T-squared test is defined in equation 6.9. It involves the computation of differences in the sample mean vectors $(\bar{M_1} - \bar{M_2})$. It also involves a calculation of the pooled variance-covariance matrix (S_p) as defined in 6.10. The symbols n_1, n_2 refer to the size of the two partitions respectively.

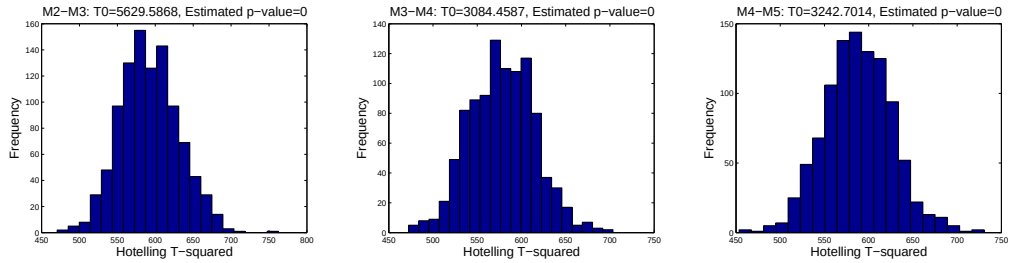
$$T^2 = \frac{n_1 n_2}{n_1 + n_2} * (\bar{M_1} - \bar{M_2}) * S_p^{-1} * (\bar{M_1} - \bar{M_2})' \tag{6.9}$$

$$S_p = \frac{(n_1 - 1) * cov(M_1) + (n_2 - 1) * cov(M_2)}{n_1 + n_2 - 2} \tag{6.10}$$

A two sample permutation test is carried out to give a simple way to compute the sampling distribution for the T-squared statistic, under the strong null hypothesis that the different partitions have absolutely no effect on the centroid. The

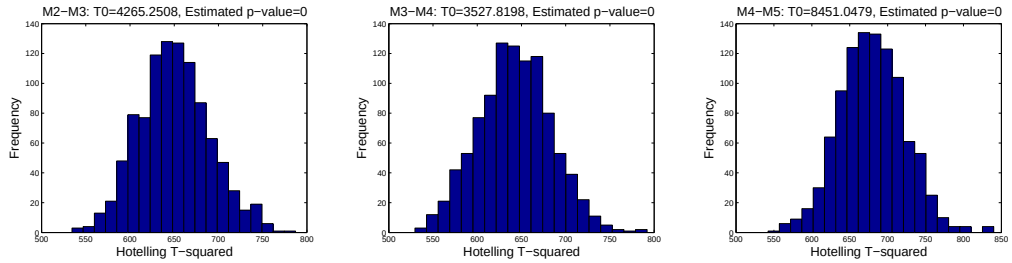
permutation principle states that the permutation distribution is an appropriate reference distribution for determining the p-value of a test and deciding whether or not a test is statistically significant. To estimate the permutation distribution of the test statistic we use M_i, M_{i+1} generated under the strong null hypothesis. In our experiment the null hypothesis claims that the centroids of M_i, M_{i+1} are equal. In the case that the null hypothesis is true, changing the exposure of M_i data points would have no effect on the outcome. By randomly shuffling the exposures many data sets are produced forming the permutation distribution of T-square statistic. In our case, 1.000 permutations are performed using 500 degrees of freedom (corresponding to 500 dimensions). Under the assumption that the null hypothesis is true the shuffled data sets should look like the original data, different otherwise. The ranking of the real test statistic T_0 among the shuffled test statistics gives a p-value. The extracted p-values, regardless the sample or the combination of partitions, indicate that the probability to meet more extreme values than the observed statistic T_0 over the original data is zero. Thus, the hypothesis of centroid equality can be safely rejected. The cluster's centroid evolves over time in a statistically significant way.

The estimated distribution of the T-squared statistic as well as the statistic of the original data (T_0) and p-value are shown in Figures 6.5, 6.6, 6.7 for the varying partitions of each sample.



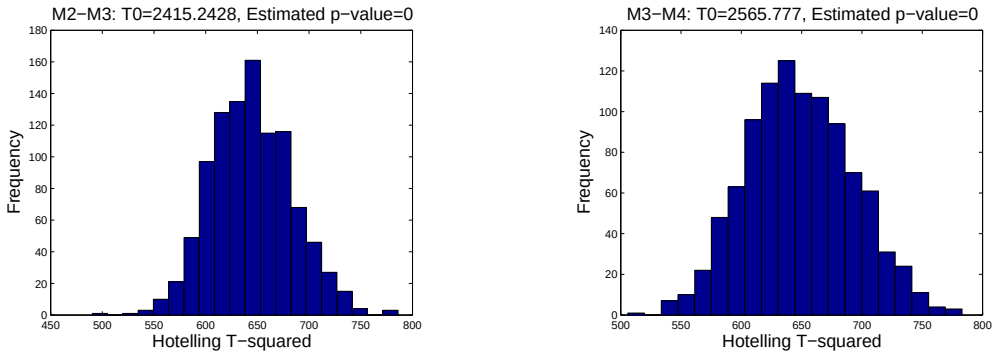
(a) (b) (c)

Figure 6.5: Hotelling's T-squared test, sample I, topic: 'Libya'



(a) (b) (c)

Figure 6.6: Hotelling's T-squared test, sample II, topic: 'Libya'



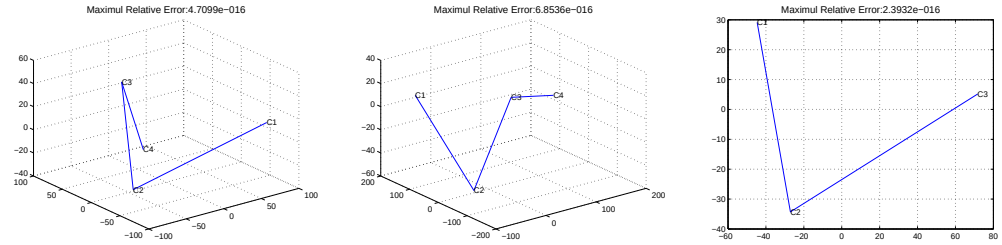
(a) (b)

Figure 6.7: Hotelling's T-squared test, sample III, topic: 'Japan'

Conceptually the statistically significant difference of the centroids of the different partitions signifies that the distances between the representative words of the different clusters increase. An increment of distance means that the weights of the words change and thus their importance evolves. Some words are now appeared more significant in the discussion while others are weaker because of no frequent occurrence. The contribution of the terms in the discussion change over time. Moreover, the movement of centroid has a great impact on the clustering algorithms. Prototypical algorithms (section 2.2) do not take into consideration a probable move of the cluster's centroid. They primarily focus on finding the nearest neighbours inside a predefined area of a circle assuming that the centroid has a constant position in space and time. Stream-oriented algorithms (section 4.1) adjust their centroid by fading the terms weights as time passes. However, the adjustment of the centroid is a consequence of the usage of the time fading function. They do not consider the parameters affecting the evolution of the centroid that depend on the characteristics of the stream.

In order to better understand the movement of the centroid we plot in three dimensions (Figure 6.8) the centroids of the varying partitions (i.e. $c_1, c_2, ..$) and lines among them to visualize the trajectory. The reduction of the multidimensional centroid into 3-D was achieved by performing a classical multidimensional scaling [78] producing a low maximum relative error. On the other hand, the reduction of the clusters points into 3-D produces a high relative error revealing the weakness of the method to find a good low-dimensional reconstruction. Thus, we experience a lack of getting a sense of how near or far points are from each other and from centroid.

The study of the centroid inside the count-based sliding window model of t_{step} equals to N reveals a shift of topic interest. Nevertheless, the experiment gives no intuition on when this evolution happens. The temporal parameter can be determined in two ways; by finding the minimum tweets needed to bring the evolution of centroid and discover the corresponding best time window. Each of the two perspectives are discussed.



(a) Sample I (b) Sample II (c) Sample III

Figure 6.8: Centroid trajectory in 3D of the three samples

In the next step, the Hotelling's T-squared statistical test is performed using sliding windows of size N=2.000 tweets for various values of slip step n_{step} and 1.000 permutations. The first sliding window is utilized to determine the terms of the vocabulary. Inside the rest sliding windows the weights of the top 500 words are recomputed. The table 6.2 summarizes the results for the set of the three samples when $n_{step} >= 200$. The table demonstrates the range of the p-values extracted from the execution of the statistical test for each two sliding windows.

	Sample I	Sample II	Sample III
$n_{step} = 200$	1	1	[0, 1]
$n_{step} = 300$	[0.694, 1]	[0.994, 1]	[0, 1]
$n_{step} = 400$	[0, 0.976]	[0, 1]	[0, 0.994]
$n_{step} = 500$	0	[0, 0.178]	[0, 0.356]
$n_{step} \geq 600$	0	0	0

Table 6.2: Hotelling p-values for different values of n_{step}

For the first sample we note that for less than 200 new added tweets in the sliding window the centroid has no statistically significant change (p-value=1). The new tweets are slightly affecting the position of the centroid and the evolution of the vocabulary. On the other hand, the increase of the new incoming tweets to 300 reduces the certainty of unequal centroids indicating a higher affection to the centroid evolution. Nevertheless, the centroids still appear equals until the increase of n_{step} to 400. This is a turning point as some pairs of partitions appear with significant different centroids and some others with equal. Specifically, four out of twelve pairs of partitions appear with statistically significant difference of centroids while the rest present equality. For n_{step} greater than 500 the difference among centroids becomes permanent.

Almost the same behaviour is met in the second sample. For the values of n_{step} equal to 400 or 500 there exist partitions of p-value below the statistical significant level of 0.05. Specifically, for $n_{step} = 400$ only two pairs of partitions

appear evolution of centroid. This number increases to eight out of nine pairs of partitions with p-value < 0.05 for $n_{step} = 500$. The stabilization of p-value to zero comes when n_{step} is greater or equal than 600. The growing number of pairs with p-value < 0.05 indicates that increasing the number of incoming tweets and thus the amount of information leads to identification of centroid evolution. Particularly, some pairs of sliding windows appear to be more sensitive to the evolution of the centroid and they have already adapted the changes of the vocabulary predicting the upcoming evolution.

The third sample has statistically significant different centroids for $n_{step} > 600$ and for some only pairs of sliding windows where n_{step} is less or equal to 500. Equal centroids are met only for small values of n_{step}, less or equal to 50. The third sample concerns the topic Japan where the evolution of centroid is observed in some windows even with the addition of very few points.

An interesting aspect of the count-based window model is that each window corresponds to different time interval spanning from few hours to several weeks. For instance, the first sample with $n_{step} = 500$ is divided into time windows of minimum two days to maximum nineteen days. The new 500 incoming tweets correspond to an interval of five hours to six days. Thus, the percentage of fresh content inside the window that forces the centroid's evolution varies in 6-38%. Respectively, the time interval of the second sample for $n_{step} = 600$ is 1.5 hour to one day with the duration of the new content being 20 minutes to 15 hours (19-86% of fresh tweets). Last but not least, the third sample of $n_{step} = 600$ spans from two to four days with the new content between 14-42 hours long (27-42% of freshness).

To conclude the count-based sliding window model reveals that there is a minimum number (threshold) of tweets that bring the evolution of the centroid. The right selection of the slip step size (n_{step}) is a critical parameter for quickly detecting the centroid's movement. On the other hand, a poor value of that threshold can lead to misleading assumptions. The selection of the best value for n_{step} depends on the parameters affecting the evolution of the centroid, which is an issue under consideration. In an intuitive and theoretical level, among the factors that affect the evolution of the centroid may be the number of users talking about a topic, as well as how many new events arise contributing to the discussion. The dispersion of a topic (local or global interest) may accelerate or slow down the evolution of the vocabulary. Furthermore, the existence of a high number of re-tweets appearing in different windows may delay the evolution. In order to prove our assumptions, it is necessary to examine all the possible parameters of affecting the centroid in a more systematic way. This perspective is beyond the scope of this thesis and is an open issue for future work.

A main advantage of the count-based sliding window model is that it allows the processing of the data in batches and gives memory and time guarantees. It is a convenient technique for the clustering algorithm especially when dealing with on-line massive data.

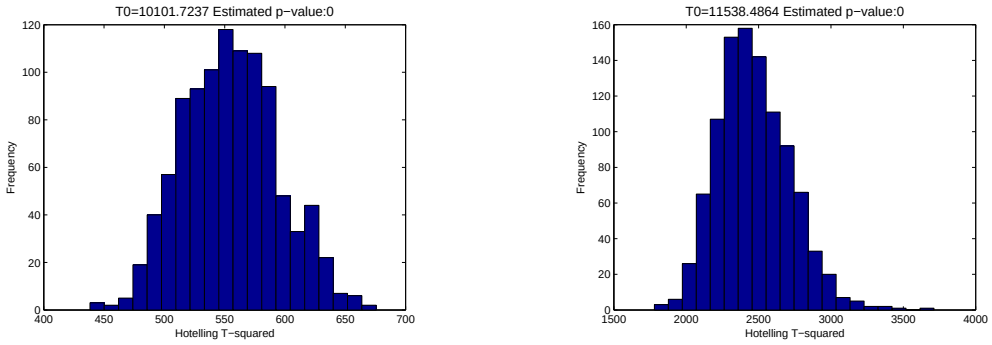
In the next subsection we experiment on the time-based sliding windows and discuss the results of the centroid's evolution statistical tests.

Time-Based Sliding Window

Assume that a data stream consists of a set of multidimensional points $p_1, ..., p_i, ...$ arriving at time stamps $t_1, ..., t_i, ...$ and $p_i = (p_1^i ... p_d^i)$. In our time window model, the window M_i starts at time t_i and ends at time t_j so that the time interval $t_j - t_i$ of the active points inside the window corresponds to a constant value Δt at any time. The points that exceed the fixed length of time window are deleted from the memory. We introduce a slip step t_{step} as a time interval during which new points are added to the window and the oldest are removed, so that the time interval of the window remains constant.

In order to detect the evolution of centroid over time and extract a result of the minimum time interval needed for the evolution, the Hotelling T-squared statistical test is applied over the three samples using the time-based sliding window model. First, we assign t_{step} equal to Δt and use the first time window of each sample for the initialization of the vocabulary. The top 500 words, based on the document frequency metric (df_{w_i}), are extracted. Inside the rest time windows the weights of the top 500 words are reassigned. For each sample, the time interval is selected based on an intuitive trade-off between the time period of the collection and its arrival rate.

For the first sample a period of one month is used to construct the time windows. The number of tweets in each time window differs based on the topic arrival rate. For example, $|M_1| = 2.250, |M_2| = 6.696, |M_3| = 660$ and $|M_4| = 165$. The first window M_1 is used to determine the vocabulary, by extracting the top 500 words. Figures 6.9 show that there is a statistically significant difference of the centroids for $\Delta t = 1 month$. The probability to meet the observed statistic T_0 in the estimated distribution is zero. Thus, a collection of one month data leads to the evolution of the vocabulary and conceptually to a statistically significant topic shifting. The focus of the discussion is changing with Twitter users differentiating their vocabulary and arguments.



(a) (b)

Figure 6.9: Hotelling's T-squared test over time windows, sample I

The second sample is selected during the bursty days ($\pm 1day$ from the day of the highest arrival rate) of the topic 'Libya' and is divided into time windows of $\Delta t = 1day$ with size $|M_1| = 4.197, |M_2| = 3.133$ and $|M_3| = 1.557$. The first time window M_1 is again used to determine the vocabulary. The statistical comparison of the rest two time windows (Figure 6.10) shows a statistically significant change of centroid, proving that for the given sample only one day's data are sufficient for the evolution of the centroid and thus of the vocabulary. The topic of conversation changes in a rapid way, from one day to another, with users discussing the breaking news of the topic and contributing to the discussion with new facts and comments.

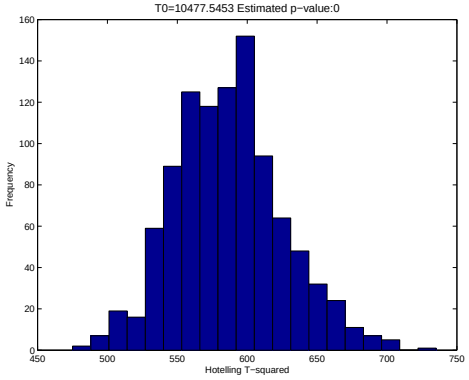


Figure 6.10: Hotelling's T-squared test over time windows, sample II

The third sample spans a period of 13 days and is separated into windows of $\Delta t = 1day$ and size in the range [71, 996]. Figure 6.11 illustrates the statistic distribution of the different windows for sample III. The Hotelling's T-squared statistic test results in significant evolution of the cluster's centroid for each two sequential windows. Although the topic is not collected during its bursty period it also appears a quick topic conversation shifting.

The experiments so far clarify a movement of centroid proving that a statistically significant evolution is observed from one time window to another. In order to obtain a better understanding on the minimum time needed for detecting the evolution we tune the values of the slip step (t_{step}) updating the contents of the window with new incoming tweets of different time duration.

We perform the Hotelling T-squared test using a constant time window (Δt) for each sample. In particular, for the first sample the size corresponds to one month and for the rest two samples to one day, as defined in the previous experiment. The varying values of t_{step} are shown in Table 6.3.

The thematic shift of the first sample can be observed earlier in time compared to the previous experiment, with the addition of fifteen days data instead of a month. The Twitterers seem to discuss about the topic for a long period by repeating part of the initial arguments and opinions. The vocabulary of the first month is almost static, appearing great similarity with the discussion of the up-

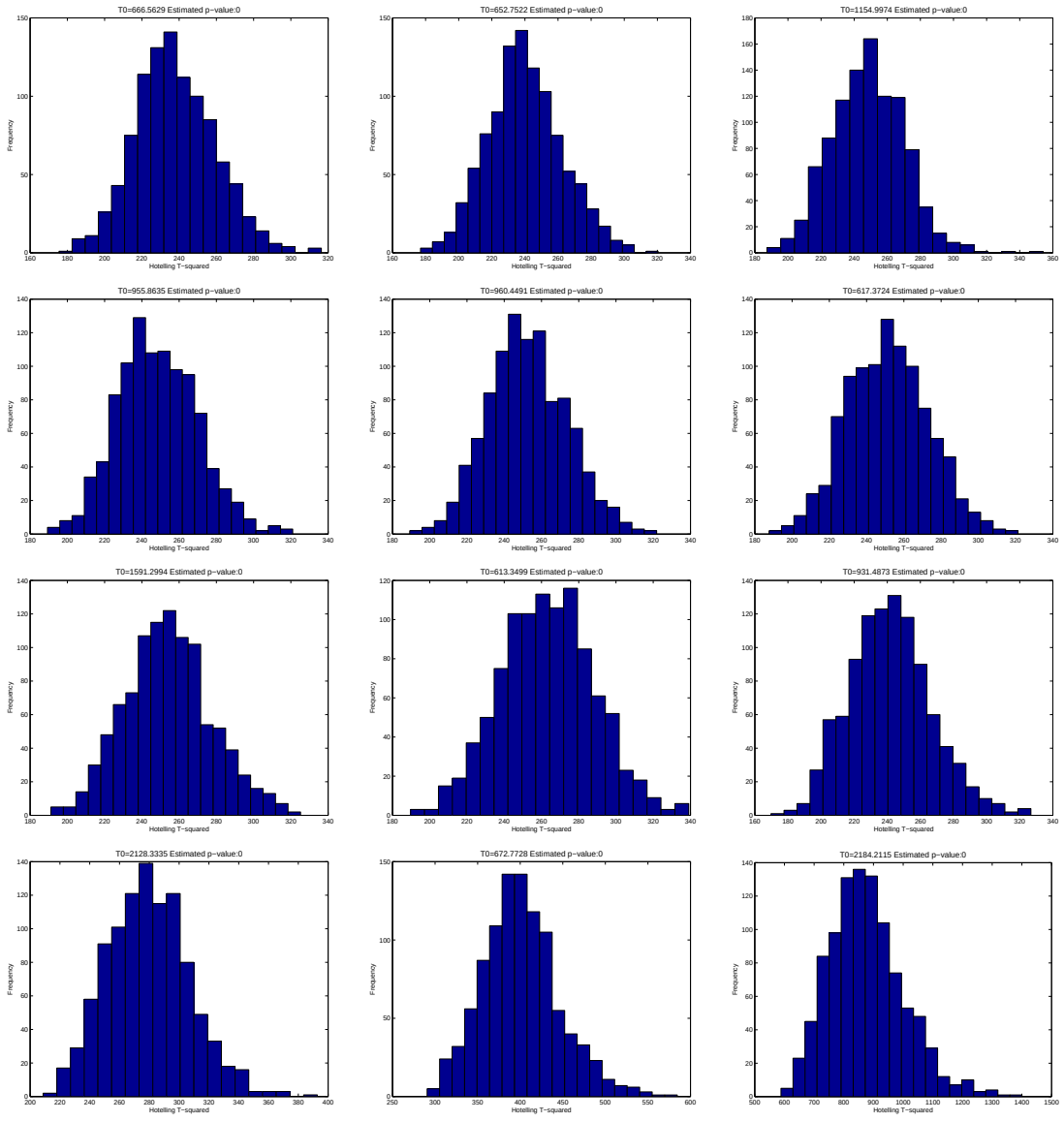


Figure 6.11: Hotelling's T-squared test over time windows, sample III

coming tweets of the first half of the next month. The topic has no statistically significant evolution and no new events appear in reality. Nevertheless, the vocabulary is inevitably differentiated when comparing the difference between periods of high time distance.

For the second sample the evolution is captured even with $t_{step} = 12h$ revealing that in a period of two days the users create and add new content to the network every few hours affecting the initial subject. The topic seems to be particularly active with new events being added to the topic. The users modify their vocabulary making the vocabulary of the previous few hours unrepresentative.

The third sample follows the same behaviour with the second but with lower centroid evolution rate. Although this sample is not collected around its bursty period it seems to be quickly influenced by the real-life breaking news. Twitters adapt their terminology to the new events quickly affecting the trajectory of the centroid. It is already observed that after the earthquake and tsunami struck in Japan many citizens used the Twitter as a news media or to express their feelings.

	Sample I	Sample II	Sample III
$t_{step} = 1h$		[0,1]	1
$t_{step} = 12h$	1	0	[0, 0.995]
$t_{step} = 24h$	[0, 1]	0	0
$t_{step} = 1week$	[0, 1]		
$t_{step} = 15days$	0		
$t_{step} = 1month$	0		

Table 6.3: Hotelling p-values for different values of t_{step}

To conclude, tracking the trajectory of the centroid extracts valuable knowledge on the evolution of a topic. It provides a way to infer the stagnation or evolution of a topic, based on the users discussion and variation of their sayings. In addition, it provides the clustering algorithms with an additional knowledge in order to adjust their parameters and capture the trend of the stream.

Furthermore, the usage of the time-based sliding window model gives the advantage of tuning the data freshness and study the behaviour of varying size windows. The duration of the slip step of the window t_{step} is a critical parameter for the model as it sets the time sensitivity for the evolution detection. A low time interval can lead to the ignorance of the centroid movement and to the weakness of a clustering algorithm to absorb the evolution and adjust its own cluster centroid. The failure of the clustering algorithm to track the centroid trajectory reduces the quality of the results. The algorithm fails to detect a new subtopic and its relevant content as well as to adapt to the evolving content. A high time interval increases the sensitivity of the centroid tracking, but can not capture the intermediate steps that led to the displacement. From the perspective of a clustering algorithm, with no concern of capturing the evolution, a high value of t_{step} will lead to the creation

of new clusters without aware of the common topic.

Dynamic Weighting Model

The experiments performed so far suppose a static environment where the most representative features are extracted from an a-priori known corpus. In addition, the weights of the terms do not reflect the stream evolution as they do not change dynamically. In fact, in a realistic scenario the tweets arrive dynamically and their importance change over time. A key challenging issue of mining such social textual streams is how to analyse the real-time distributed messages and extract significant features of them in a dynamic environment.

In order to tackle these challenges we performed the Hotelling's T-squared statistic test using a dynamic weighting scheme, proposed in [21], with time-based sliding windows. The intuition behind bursT weighting method is to penalize the uninformative words that occur frequently with low burstiness (i.e. haha, lol) or very rarely (i.e. oral words) and highlight those that have higher burst than expectation within a certain range of document frequency.

The bursT weighting formula is presented in details in section 2.1.3 and the main equation is shown in 6.11. The $BS_{w,t}$ (Burst Score) factor captures the words burstiness and $TOP_{w,t}$ (Term Occurrence Probability) represents the probability of the word occurrence in the sliding window.

$$weight_{w,t} = BS_{w,t} * TOP_{w,t} \tag{6.11}$$

In our experimental set-up we use the 500 words, from the set of 2.000 firstly met tweets, of the top document frequency (df_{w_t}) value to initialize the vocabulary. We set the length (Δt) of the time-based sliding window to a month or day (depending on the sample) and test on varying values of the slip step t_{step}. The arrival time of the tweets are assigned with accuracy of a minute, i.e. tweets arrived with less than 60 seconds difference are assigned with the same timestamp.

Table 6.4 summarizes the range of the p-values extracted from the statistic test for all the time-based windows. We will comment on the results in reverse order of the samples appearance.

For the third sample, mentioned before for its highly evolving centroid, the experiments conclude to a non statistically significant evolution for some of the time-based sliding windows. To explain the results we need to discuss the dynamic weighting scheme. The weighting method absorbs the evolution of the centroid by penalizing the words that are no more active and increasing those that are currently in use. This behaviour causes the centroid to re-adjust its position in a statistically insignificant way. Unlike before, the centroid's stagnation is not translated as non-development of the topic but it is due to the dynamic weighting scheme. Nevertheless, there are still sliding windows with centroids that differ significantly, indicating that the dynamic weighting scheme can not fully compensate the changes and keep the centroid constant in space over time.

On the other hand, the second sample results in a statistically significant centroid inequality. The sample II has been collected during its bursty period setting the dynamic weighting scheme capable of capturing the burstiness. The first factor of bursT, $BS_{w,t}$, assumes an arrival rate of the word w at time t greater than its expectation value, which is a valid scenario of an upcoming topic. The weighting scheme tends to increase the importance of the evolving terms forcing the centroid to move.

The sample I has a time delay on detecting the statistically significant evolution of the centroid compared to the previous experiment of static weighting. Similar to the third sample, its centroid is being re-adjusted by the dynamic weighting in most of the cases.

	Sample I ($\Delta t = 1mon$)	Sample II ($\Delta t = 1day$)	Sample III ($\Delta t = 1day$)
$t_{step} = 1h$		0	[0.051, 1]
$t_{step} = 12h$	[0, 1]	0	[0, 0.997]
$t_{step} = 24h$	[0, 1]	0	[0, 0.179]
$t_{step} = 1week$	[0, 1]		
$t_{step} = 15days$	[0, 0.255]		
$t_{step} = 1month$	0		

Table 6.4: Hotelling p-values for different values of t_{step} for bursT method

Conclusions & Comparison of Window Models

Before the discussion of the cluster's shape evolution, we can summarize the discussion so far offering some valuable conclusions. Firstly, we mention that all three samples resulted in centroid movement over time. This movement indicates a shift of users interest over time expressed with the usage of different vocabulary and terminology. An unavoidable result of each conversation is the differentiation of the arguments over time and the addition of new words. The moment of the evolution and how quickly it appears is an intrinsic feature of the dataset depending on a variety of parameters that is beyond the scope of this thesis. However, the first observation is that the topics of high burstiness tend to change their representative words in a short period in order to capture the real-life breaking news.

Both of the sliding window models proved to be proper for capturing the centroid evolution. Each one is oriented in different application requirements, with the first providing memory guarantees and the second easy tuning of data freshness. Both of them need a proper selection of the slip step in order to quickly detect the centroid's movement. Furthermore, the bursT dynamic weighting scheme tends to re-adjust the terms weights of the centroid encapsulating the evolution. However, it produces centroids of statistically significant difference capturing very quickly the shift of interest.

6.2.2 Shape Evolution

Another aspect of the cluster's properties is the change of shape in time. Most of the traditional algorithms imply a structure on data. For example, k-means [1] supposes a spherical shape where points distances from centroid are minimized. The same assumption applies for DBScan algorithm [31] where the spherical clusters are synthesized into tubes of spheres. Therefore, the knowledge of cluster's shape and its dynamic evolution is a key for setting the specifications of a clustering algorithm.

The shape of the cluster gives a notion of points distance from the centroid. An almost constant shape of non significant evolution reveals a cluster where people discuss about a topic with the same interest over time using the same representative vocabulary. On the other hand, an evolving shape indicates a discrepancy between the discussed topic in the past and in current moment. An expansion of shape means that the points are getting away from the centroid and simultaneously increase their between distance. There are opinions that is far from the average and discordant views among users. A shrinkage of the cluster's shape shows that the users tend to agree with each other and are shrunk around the same vocabulary. A representative opinion expresses the majority of users. The above analysis is studied in this subsection where the shape evolves in space and time simultaneously with the centroid.

First, we study the histogram of distances from cluster's centroid for the different windows of a topic for fixed size length of 2.000 tweets. The various colors in Figure 6.12 depict the different probability distributions of the distances from centroid for every window. We observe a modification of the distribution over time for all the samples and a tendency for shrinkage around the centroid. Particularly, the first window M_2 of each topic presented in the background with red color tends to shrink over time leading to the distribution of the foreground M_5 (M_4 for sample III). The variance measure of the distributions reduces over time distinguishing the different probabilities. In the next subsection, we study the statistical importance of shape modification over the two sliding window models and the practical importance for the algorithms.

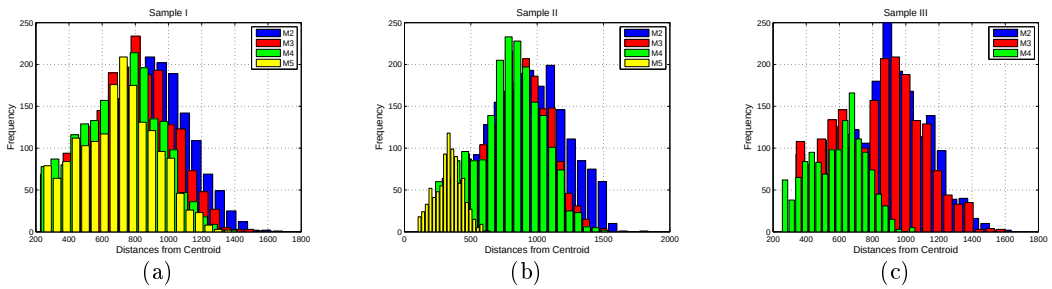


Figure 6.12: Histogram of distances from cluster's centroid over time

Count-based Sliding Window

In order to support our assumption of shape modification we divide our samples into partitions using the same method with the study of centroid (section 6.2.1) and we compute the two sample Kolmogorov-Smirnov statistic test. Kolmogorov-Smirnov statistic test compares two different distributions under the null hypothesis that they arrive from the same continuous distribution. The statistic is defined as:

$$D_{n,n'} = sup_x |F_{1,n}(x) - F_{2,n'}(x)| \tag{6.12}$$

where $F_{1,n}, F_{2,n'}$ are the empirical distribution functions of the first and the second partition respectively. The sup_x is the supremum of the set of distances. In mathematics, given a subset S of a totally or partially ordered set T, the supremum (sup) of S, if it exists, is the least element of T that is greater than or equal to every element of S.

We hypothesize that every two distributions of the varying partitions differ in a statistically significant way and thus the shape evolves in time. We use count-based sliding windows of length $\Delta t = 2.000$, cosine similarity metric and slip step n_{step} equals to Δt. Table 6.5 summarizes the observed statistics and the corresponding p-values (p_0). The values of p_0 is always above the statistical significance level ($>$ 0.05) showing a statistically significant change in shape of all the windows of each sample.

The statistically significant change of shapes over time indicates the plurality of users opinions. As time passes users change their point of view. Sometimes they tend to agree with one representative opinion forcing the data points to approach the centroid of the cluster. They also tend to agree with each other with the majority of people embrace a common opinion and share a common feeling about a topic. This behaviour makes the data points of the cluster to approach each other and the shape to shrink. On the other hand, an expansion of the cluster depicts a topic of highly varying opinions with people disagree with each other.

	Sample I	Sample II	Sample III
$M_2 - M_3$	$T_0 = 0.3175$ $p_0 = 1.1395e - 88$	$T_0 = 0.479000$ $p_0 = 2.8296e{-}201$	$T_0 = 0.094547$ $p_0 = 3.0178e - 8$
$M_3 - M_4$	$T_0 = 0.2255$ $p_0 = 6.1300e - 45$	$T_0 = 0.2125$ $p_0 = 5.9143e - 40$	$T_0 = 0.300830$ $p_0 = 7.4871e - 65$
$M_4 - M_5$	$T_0 = 0.288127$ $p_0 = 1.1093e - 68$	$T_0 = 0.149429$ $p_0 = 1.9146e - 12$	

Table 6.5: Kolmogorov-Smirnov test results for the count-based window model

So far the study of the cluster's shape proved to result in cluster evolution.

However, two issues remain to be answered; the first deals with the time on which the evolution happened and the second the increase or reduction of shape's density.

In the purpose of studying the time parameter of the evolution we test our hypothesis on different values of slip step n_{step}. We apply the two-sample Kolmogorov-Smirnov test with $\Delta t = 2.000$. Table 6.6 summarizes the range of the p-values of every window for each value of n_{step}.

For the first sample, when the addition of new points are greater than or equal to 400, the p-values are below the significance level of 0.05 and thus the shape evolves in a statistically significant way. For less than 400 new tweets there are windows that their shape statistically differs and windows of the same shape distribution. The addition of 400 new points causes the evolution of shape which is earlier detected that the evolution of centroid. The Twitter users tend to differentiate their opinions over time and then affecting the representative words of the centroid.

The second sample evolves in shape with the addition of 400 new points but has no statistically significant evolution with the addition of less than 400. Furthermore, the addition of 500 new points results in a statistically significant evolution of all the windows except one. The same points in the window of $n_{step} = 400$ also received a relative high p-value but below the significance level.

The partitions of the third sample evolve in shape with the addition of more than 400 new points. It is worth mentioning that the evolution of shape happens earlier in time than the movement of centroid. The cluster first changes its shape and then moves its centroid.

	Sample I		Sample II		Sample III	
$n_{step} = 200$	$[1.11767e$ $15, 0.15762]$	–	$[5.44436e$ $20, 0.860046]$	–	$[4.48313e$ $244, 0.97717]$	–
$n_{step} = 300$	$[2.56614e$ $28, 0.100643]$	–	$[6.93591e$ $14, 0.194395]$	–	$[1.10690e$ $268, 0.573835]$	–
$n_{step} = 400$	$[3.47571e$ $26, 0.0032002]$	–	$[6.13003e$ $45, 0.00500738]$	–	$[0, 0.0272378]$	
$n_{step} = 500$	$[5.00745e$ $32, 9.37744e-8]$	–	$[7.34041e$ $32, 0.344326]$	–	$[2.07551e$ $285, 0.00141138]$	–
$n_{step} = 600$			$[2.7981e$ $25, 4.2640e-8]$	–	$[5.7879e$ $312, 5.3425e-5]$	–

Table 6.6: Kolmogorov-Smirnov p-values for different values of n_{step}

Time-based Sliding Window

The window model used in subsection 6.2.1 is also applied for the study of shape evolution. The selected time window's length is one month, one day and one day for each sample respectively. In the simple concept where the slip step t_{step} is equal to the window length $\Delta t = 2.000$, we observe (Table 6.7) that the shape evolves no matter the sample or the window.

	Sample I	Sample II	Sample III
$M_2 - M_3$	$T_0 = 0.2919$ $p_0 = 2.41244e{-}45$	$T_0 = 0.2023$ $p_0 = 1.1474e - 37$	*Less than 0.05 for all partitions*
$M_3 - M_4$	$T_0 = 0.2621$ $p_0 = 1.75698e - 8$		

Table 6.7: Kolmogorov-Smirnov test results for the time-based window model

In order to obtain a better understanding we present in Table 6.8 the range of p-values for the various time-based windows applying different values to t_{step}. Finally, we conclude that the evolution of shape is detected earlier in time in some cases than that of the centroid.

	Sample I	Sample II	Sample III
$t_{step} = 1h$		$[4.05784e - 40, 0.998776]$	$[8.89429e - 14, 1]$
$t_{step} = 12h$	$[2.42101e - 25, 1]$	$[1.62261e - 227, 7.33766e{-}11]$	$[1.03743e - 20, 0.226018]$
$t_{step} = 24h$	$[1.92449e - 47, 1]$	$1.16944e - 246$	$[9.67800e - 25, 3.22345e - 3]$
$t_{step} = 1week$	$[1.97626e - 323, 0.465112]$		
$t_{step} = 15days$	$[0, 2.43556e - 4]$		
$t_{step} = 1month$	$[7.59122e - 42, 2.52365e - 24]$		

Table 6.8: Kolmogorov-Smirnov p-values for different values of t_{step}

The first sample detects the variation of users opinions with the addition of tweets corresponding to one week while the centroid shift is captured with the

addition of fifteen days data. For the rest two samples the detection of shape is achieved at the same slip step with that of the centroid.

Dynamic Weighting Model

The dynamic weighting model of bursT is also used for testing our initial hypothesis of shape equality over time where the terms are weighted based on their arrival rate and probability occurrence. The size window length is also set to $\Delta t = 2000$ as described in the experimental settings of 6.2.1.

The dynamic weighting re-positions the points of the cluster in a way that the shape changes in an insignificant way in some of the cases. Specifically, the evolution of centroid for the sample I is captured when $t_{step} = 15days$ appearing the same behaviour with the static weighting scheme of $tf \cdot idf_2$. For the sample II the addition of twelve hours tweets causes modification of the shape verifying the results of the previous experiment. On the contrast, no modification of shape is observed for the third sample which was captured early with the usage of static weighting.

	Sample I	Sample II	Sample III
$t_{step} = 1h$		$[4.00706e - 41, 0.104821]$	$[1.31377e - 3, 0.999954]$
$t_{step} = 12h$	$[4.19649e - 30, 0.999125]$	$[2.96414e - 42, 8.27652e - 5]$	$[8.35308e - 6, 0.119138]$
$t_{step} = 24h$	$[8.32224e - 26, 0.999966]$	$8.24307e - 111$	$[1.39173e - 3, 0.533157]$
$t_{step} = 1week$	$[1.47527e - 19, 0.131928]$		
$t_{step} = 15days$	$[2.1994e - 21, 9.64815e - 4]$		
$t_{step} = 1month$	$[6.9765e - 18, 5.1034e - 13]$		

Table 6.9: Kolmogorov-Smirnov p-values for different values of t_{step} for bursT method

6.2.3 Shape Recognition

By studying the clusters in different windows, we have seen that their centroid is moving in space and simultaneously their shape is transformed. The next issue being under consideration deals with clusters' density and their tendency to shrink or expand. The discovery of cluster's shape tendency and characteristics of change will give a better understanding of the cluster's behaviour.

The figure 6.13 shows the variation of distances from centroid for the different partitions of fixed size windows ($\Delta t = n_{step} = 2000$) for each sample. The vocabulary is extracted as the top 500 terms of the first test-window, which is also depicted in the chart. Hence, in the first time period of the first two samples (M_1, M_2) the clusters tend to expand while after the second sliding window the clusters shrink over time reducing their volume. In the first period tweets are diverse with users expressing higher discrepancy of opinions compared with the latest periods. On the other hand, within the following sliding windows the points are approaching each other inside the cluster and move towards their centroid. A possible explanation of this behaviour outlines that tweets appear a great similarity among each other with common vocabulary and words of the same contribution. Thus, as time progresses Twitters shape a common opinion and their arguments are converging.

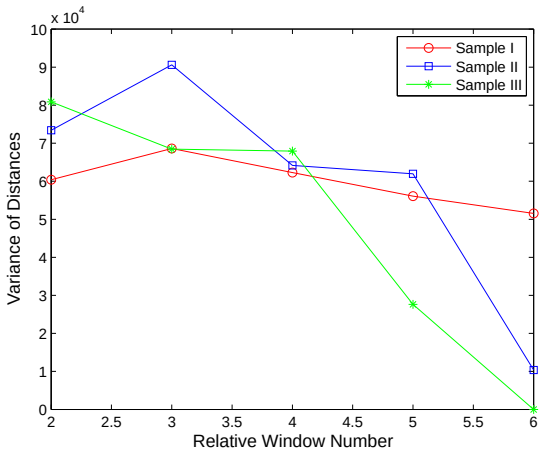


Figure 6.13: Variance of distances from cluster's centroid, constant FS

A significant observation in the behaviour of the clusters over time is the existence of a fixed feature space. The aforementioned experiments are performed over points that their weights concern only the terms belonging to the pre-defined feature space of the first test data window (M_1). The new terms arriving over time are not encompassed in the centroid and are not considered in the evolution. Thus, the shrinkage of the clusters' shape might have been caused due the lack of representative terms. So far, we can claim that the users tend to experience the usage of a core vocabulary but we need to test if the consideration of the new upcoming terms within each sliding window causes a probable expand of the shape.

In the purpose of studying the behaviour of the cluster's shape for a non constant feature space, we extract the top 500 terms based on the highest df_{w_i} metric for each sliding window. Figure 6.14 illustrates that even for an adjustable feature space the behaviour of the clusters is similar. Therefore, we can state that users tend to agree with each other as time progresses, sharing a common opinion.

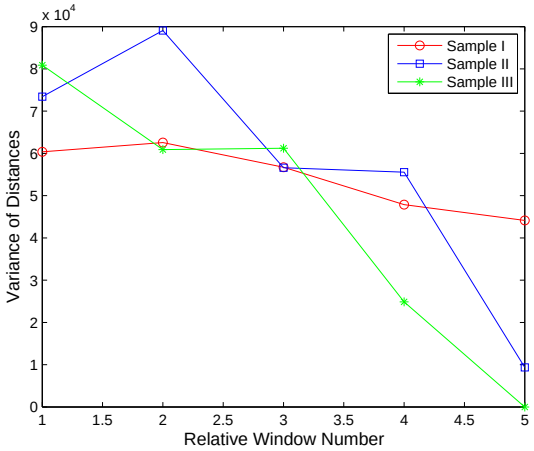


Figure 6.14: Variance of distances from cluster's centroid, non constant FS

Furthermore, the usage of an adaptable feature space causes a stronger shrinkage of clusters' shape than that of a static feature space. Figure 6.15 facilitates the assessment of this observation as it depicts the variance of points from centroid over time for the constant and for the adjustable feature space of each sample. We mention that the curve of the points variance for the non-constant feature space is always above the curve of the constant feature space for all the samples. This observation denotes that reconsidering the feature space and thus resulting in more representative terms, we result in more compact clusters and we can safely predict the convergence of users opinions.

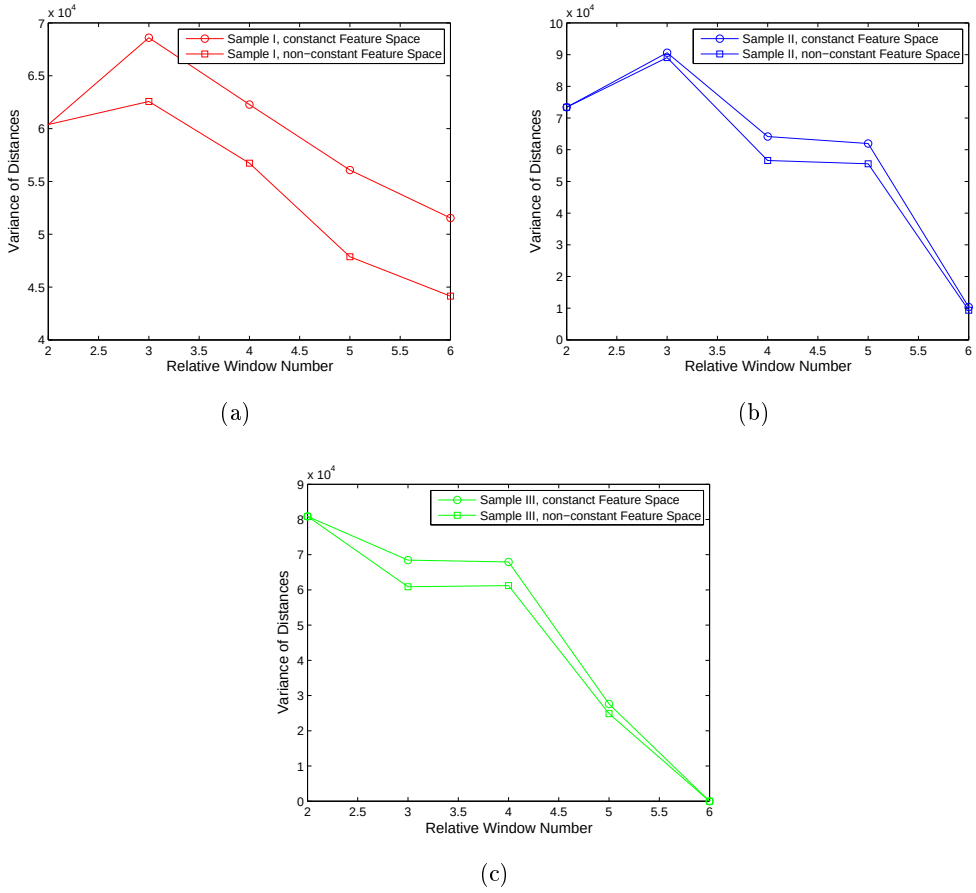


Figure 6.15: Points variance from centroid for constant and non-constant feature space

Chapter 7

Conclusions & Future Work

In this thesis we have tackled the problem of providing a methodological framework for statistical analysis of User Generated Content (UGC) produced in social media that considers the peculiarities of the social text streams exhibited in reality. The low quality posts, the heterogeneous topics that range from personal stories to breaking news and the evolving nature of the social text stream are encompassed in the peculiarities, deteriorating the poor clustering quality results of several well-known algorithms. In particular, within the scope of our dataset we have evidence for each of the following conclusions.

- The widely used $tf \cdot idf$ weighting scheme for document clustering exhibits limitations over the short posts of social streams with the factor of tf ranging in [0, 3]. We showed that using a variation of this scheme, the $tf \cdot idf_2$, we result in a weighting representation of better discriminative ability among the clusters. Furthermore, the distance metric of cosine similarity has a better behaviour in a high dimensional space than other distance metrics (e.g. euclidean), as it considers the similarity on the existed terms and not on the absence of them.

- The centroid of the cluster within a pre-defined tag evolves in a multi-dimensional space over time in a statistically significant way, illustrating a modification of the representative vocabulary and thus a shift of users' interest in the discussed topic. Several clustering algorithms (e.g. denStream) reconsider the position of the centroid by decaying its terms over time. Nevertheless, the parameters affecting the movement of centroid are not limited in the time progression but are intrinsic characteristics of the topic. In particular, topics of high burstiness in terms tend to evolve earlier in time as the importance of their terms significantly change forcing the vocabulary to evolve.

- A dynamic weighting scheme (bursT) was used to capture the bursty nature of the stream, favouring the terms of higher arrival rate than expectation. The experiments depicted that when dealing with a bursty topic using the

bursT method the movement of centroid is captured earlier in time compared with the static weighting. On the other hand, the evolution of centroid for topics of lower burstiness spanning in a longer period is not reflected from the dynamic scheme. Instead, it is adjusting the weights of the clusters' points in a way that it absorbs the evolution of the vocabulary.

- The evolution of centroid utilizing different techniques of sliding windows (count-based and time-based) benefits the clustering procedure by either providing memory guarantees or easy tuning of data freshness. We showed that there exists a threshold either referring to the number of new incoming tweets or to a time parameter above which the evolution is detected. The selection of the threshold is a critical issue for the algorithms as a low value can lead to the evolution ignorance and thus to no re-adjustment of the centroid, while a high value to the creation of too many clusters of the same topic without thematic relation among them.

- The shape of the cluster tends to evolve statistically significant over time indicating the users' opinion convergence or discrepancy within a topic. We showed that all the clusters, regardless the topic, are shrinking around the centroid as time progresses. The reduction of clusters' density is observed even with an adjustable feature space over the sliding windows, illustrating the propensity of the users to converge around a public opinion and finally agree with each other.

Several directions for future work are considered in order to support the analysis of social text streams and understand the behaviour of the existed social text stream-oriented clustering algorithms.

The first research aspect deals with the generalization of the aforementioned statements beyond the narrow limits of our collection of the pre-defined tags. The scalability of the extracted results can be provided by the analysis of a broader set of data arriving from the UGC stream.

Another interesting direction for future exploration is the illustration of several clustering algorithms' weaknesses to adjust the analysed clusters to the peculiarities of social text streams and improve their clustering results. Our preliminary findings on the clustering algorithms, motivated us to study the nature of the stream, can be further examined to shape a better understanding of the algorithms' limitations.

As a final but equally important aspect, we consider the exploitation of the extracted knowledge by devising a specialized machine learning algorithm that will be able to adjust its centroid and shape on-the-fly adapted to the dynamic nature of the social stream.

Bibliography

[1] J. A. Hartigan and M. A. Wong, "A K-means clustering algorithm," *Applied Statistics*, vol. 28, pp. 100–108, 1979.

[2] M. Zimmermann, I. Ntoutsi, Z. F. Siddiqui, M. Spiliopoulou, and H.-P. Kriegel, "Discovering global and local bursts in a stream of news," in *Proceedings of the 27th Annual ACM Symposium on Applied Computing*, ser. SAC '12. New York, NY, USA: ACM, 2012, pp. 807–812. [Online]. Available: http://doi.acm.org/10.1145/2245276.2245433

[3] Y. Zhao and G. Karypis, "Criterion functions for document clustering: Experiments and analysis," Tech. Rep., 2002.

[4] J. Weng, Y. Yao, E. Leonardi, F. Lee, and B.-s. Lee, "Event detection in twitter," *Development*, no. 98, pp. 401–408, 2011. [Online]. Available: http://www.aaai.org/ocs/index.php/ICWSM/ICWSM11/paper/download/2767/3299

[5] C. D. Manning, P. Raghavan, and H. Schtze, *Introduction to Information Retrieval*. New York, NY, USA: Cambridge University Press, 2008. [Online]. Available: http://portal.acm.org/citation.cfm?id=1394399

[6] C. C. Aggarwal and C. Zhai, Eds., *Mining Text Data*. Springer, 2012.

[7] J. B. Lovins, "Development of a Stemming Algorithm." Jun. 1968. [Online]. Available: http://stinet.dtic.mil/oai/oai?verb=getRecord&metadataPrefix=html&identifier=AD0735504

[8] M. F. Porter, "Readings in information retrieval," K. Sparck Jones and P. Willett, Eds. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1997, ch. An algorithm for suffix stripping, pp. 313–316. [Online]. Available: http://dl.acm.org/citation.cfm?id=275537.275705

[9] C. Lioma and I. Ounis, "Light syntactically-based index pruning for information retrieval," *Lecture Notes in Computer Science*, vol. 4425, pp. 88–100, 2007. [Online]. Available: http://eprints.gla.ac.uk/3769/

[10] G. Kumaran and J. Allan, "Text classification and named entities for new event detection," 2004.

[11] W. Lam, H. M. L. Meng, K. L. Wong, and J. C. H. Yen, "Using contextual analysis for news event detection," *International Journal on Intelligent Systems*, 2001.

[12] J. Makkonen, H. Ahonen-myka, and M. Salmenkivi, "Applying semantic classes in event detection and tracking," in *In: Proc. International Conference on Natural Language Processing (ICON'02*, 2002, pp. 175–183.

[13] T. H. Cao, T. M. Tang, and C. K. Chau, "Text clustering with named entities: A model, experimentation and realization," in *Data Mining: Foundations and Intelligent Paradigms*, ser. Intelligent Systems Reference Library, D. E. Holmes and L. C. Jain, Eds. Springer Berlin Heidelberg, 2012, vol. 23, pp. 267–287.

[14] M.-A. Jashki, M. Makki, E. Bagheri, and A. A. Ghorbani, "An iterative hybrid filter-wrapper approach to feature selection for document clustering," in *Proceedings of the 22nd Canadian Conference on Artificial Intelligence: Advances in Artificial Intelligence*, ser. Canadian AI '09. Berlin, Heidelberg: Springer-Verlag, 2009, pp. 74–85. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-01818-3_10

[15] J. Yang and Z. Ma, "Document clustering based on mutual information and pca subspace," in *Artificial Intelligence, Management Science and Electronic Commerce (AIMSEC), 2011 2nd International Conference on*, aug. 2011, pp. 2983 –2986.

[16] I. Fodor, "A survey of dimension reduction techniques," Tech. Rep., 2002.

[17] M. A. Carreira-Perpinan, "A review of dimension reduction techniques," 1997.

[18] G. Kumaran and J. Allan, "Using names and topics for new event detection," in *Proc. of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, ser. HLT '05. Stroudsburg, PA, USA: Association for Computational Linguistics, 2005, pp. 121–128. [Online]. Available: http://dx.doi.org/10.3115/1220575.1220591

[19] T. Brants, F. Chen, and A. Farahat, "A system for new event detection," in *Proc. of the 26th annual international ACM SIGIR conference on Research and development in information retrieval*, ser. SIGIR '03. New York, NY, USA: ACM, 2003, pp. 330–337. [Online]. Available: http://doi.acm.org/10.1145/860435.860495

[20] Y. Yang, T. Pierce, and J. Carbonell, "A study of retrospective and online event detection," in *Proc. of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, ser. SIGIR '98. New York, NY, USA: ACM, 1998, pp. 28–36. [Online]. Available: http://doi.acm.org/10.1145/290941.290953

[21] C.-H. Lee, C.-H. Wu, and T.-F. Chien, "Burst: a dynamic term weighting scheme for mining microblogging messages," in *Proc. of the 8th international conference on Advances in neural networks - Volume Part III*, ser. ISNN'11. Berlin, Heidelberg: Springer-Verlag, 2011, pp. 548–557. [Online]. Available: http://dl.acm.org/citation.cfm?id=2009463.2009531

[22] P. Berkhin, "Survey of clustering data mining techniques," Tech. Rep., 2002.

[23] T. T. Win and L. Mon, "Document clustering by fuzzy c-mean algorithm," in *Advanced Computer Control (ICACC), 2010 2nd International Conference on*, vol. 1, march 2010, pp. 239 –242.

[24] V. Singh, N. Tiwari, and S. Garg, "Document clustering using k-means, heuristic k-means and fuzzy c-means," in *Computational Intelligence and Communication Networks (CICN), 2011 International Conference on*, oct. 2011, pp. 297 –301.

[25] M. Steinbach, G. Karypis, and V. Kumar, "A comparison of document clustering techniques," in *In KDD Workshop on Text Mining*, 2000.

[26] B. C. Fung, K. Wang, and M. Ester, "Hierarchical document clustering using frequent itemsets," in *IN PROC. SIAM INTERNATIONAL CONFERENCE ON DATA MINING 2003 (SDM 2003*, 2003.

[27] L. W. Jian-Suo Xu, "Tcblht: A new method of hierarchical text clustering," 2005.

[28] R. Sibson, "SLINK: An optimally efficient algorithm for the single-link cluster method." [Online]. Available: http://www.cs.gsu.edu/~{}wkim/index_files/papers/sibson.pdf

[29] D. Defays, "An efficient algorithm for a complete link method," *The Computer Journal (British Computer Society)*, 1977.

[30] Y. Zhao and G. Karypis, "Evaluation of hierarchical clustering algorithms for document datasets," in *Proc. of the eleventh international conference on Information and knowledge management*, ser. CIKM '02. New York, NY, USA: ACM, 2002, pp. 515–524. [Online]. Available: http://doi.acm.org/10.1145/584792.584877

[31] M. Ester, H. peter Kriegel, J. S, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise." AAAI Press, 1996, pp. 226–231.

[32] M. Ankerst, M. M. Breunig, H. peter Kriegel, and J. Sander, "Optics: Ordering points to identify the clustering structure." ACM Press, 1999, pp. 49–60.

[33] A. M. Kaplan and M. Haenlein, "Users of the world, unite! The challenges and opportunities of Social Media," *Business Horizons*, 2010. [Online]. Available: http://openmediart.com/log/pics/sdarticle.pdf

[34] "Alexa," `http://www.alexa.com/`.

[35] "Google trends," `http://www.google.com/trends/`.

[36] "Twitpic," `http://twitpic.com/`.

[37] "Twitter api," `http://apiwiki.twitter.com/`.

[38] F. Cao, M. Ester, W. Qian, and A. Zhou, "Density-based clustering over an evolving data stream with noise," in *In 2006 SIAM Conference on Data Mining*, 2006, pp. 328–339.

[39] C. C. Aggarwal, J. Han, J. Wang, and P. S. Yu, "A framework for clustering evolving data streams," in *Proc. of the 29th international conference on Very large data bases - Volume 29*, ser. VLDB '2003. VLDB Endowment, 2003, pp. 81–92. [Online]. Available: http://portal.acm.org/citation.cfm?id=1315451.1315460

[40] "Twitter streaming api," `https://dev.twitter.com/docs/streaming-api/`.

[41] B. J. Jansen, M. Zhang, K. Sobel, and A. Chowdury, "Twitter power: Tweets as electronic word of mouth," *J. Am. Soc. Inf. Sci. Technol.*, vol. 60, pp. 2169–2188, November 2009. [Online]. Available: http://dx.doi.org/10.1002/asi.v60:11

[42] "Beyond microblogging: Conversation and collaboration via twitter," in *Proc. of the 42nd Hawaii International Conference on System Sciences*. Washington, DC, USA: IEEE Computer Society, 2009, pp. 1–10. [Online]. Available: http://portal.acm.org/citation.cfm?id=1488734.1489848

[43] D. Boyd, S. Golder, and G. Lotan, "Tweet, tweet, retweet: Conversational aspects of retweeting on twitter," in *Proc. of the 2010 43rd Hawaii International Conference on System Sciences*, ser. HICSS '10. Washington, DC, USA: IEEE Computer Society, 2010, pp. 1–10. [Online]. Available: http://dx.doi.org/10.1109/HICSS.2010.412

[44] H. Kwak, C. Lee, H. Park, and S. Moon, "What is Twitter, a social network or a news media?" in *WWW '10: Proc. of the 19th international conference on World wide web*. New York, NY, USA: ACM, 2010, pp. 591–600.

[45] M. Naaman, J. Boase, and C.-H. Lai, "Is it really about me?: message content in social awareness streams," in *Proc. of the 2010 ACM conference on Computer supported cooperative work*, ser. CSCW '10. New York, NY, USA: ACM, 2010, pp. 189–192. [Online]. Available: http://doi.acm.org/10.1145/1718918.1718953

[46] T. Zhang, R. Ramakrishnan, and M. Livny, "Birch: an efficient data clustering method for very large databases," *SIGMOD Rec.*, vol. 25, pp. 103–114, June 1996. [Online]. Available: http://doi.acm.org/10.1145/235968.233324

[47] S. Thomopoulos, D. Bougoulias, and C.-D. Wann, "Dignet: an unsupervised-learning clustering algorithm for clustering and data fusion," *Aerospace and Electronic Systems, IEEE Transactions on*, vol. 31, no. 1, pp. 21 –38, 1995.

[48] M. Bilenko, B. Kamath, and R. J. Mooney, "Adaptive blocking: Learning to scale up record linkage," *Data Mining, IEEE International Conference on*, vol. 0, pp. 87–96, 2006.

[49] B. Babcock, S. Babu, M. Datar, R. Motwani, and J. Widom, "Models and issues in data stream systems," in *Proc. of the twenty-first ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, ser. PODS '02. New York, NY, USA: ACM, 2002, pp. 1–16. [Online]. Available: http://doi.acm.org/10.1145/543613.543615

[50] M. Datar, A. Gionis, P. Indyk, and R. Motwani, "Maintaining stream statistics over sliding windows," in *SIAM Journal on Computing*, 2002, pp. 635–644.

[51] A. Zhou, F. Cao, W. Qian, and C. Jin, "Tracking clusters in evolving data streams over sliding windows," *Knowl. Inf. Syst.*, vol. 15, pp. 181–214, May 2008. [Online]. Available: http://portal.acm.org/citation.cfm?id=1388728.1388730

[52] "Streaming-data algorithms for high-quality clustering," in *Proc. of the 18th International Conference on Data Engineering*, ser. ICDE '02. Washington, DC, USA: IEEE Computer Society, 2002, pp. 685–. [Online]. Available: http://portal.acm.org/citation.cfm?id=876875.878995

[53] C. C. Aggarwal, J. Han, J. Wang, and P. S. Yu, "A framework for projected clustering of high dimensional data streams," in *Proc. of the Thirtieth international conference on Very large data bases - Volume 30*, ser. VLDB '04. VLDB Endowment, 2004, pp. 852–863. [Online]. Available: http://dl.acm.org/citation.cfm?id=1316689.1316763

[54] Y. Chen and L. Tu, "Density-based clustering for real-time stream data," in *Proc. of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, ser. KDD '07. New York, NY, USA: ACM, 2007, pp. 133–142. [Online]. Available: http://doi.acm.org/10.1145/1281192.1281210

[55] C. Ruiz, E. Menasalvas, and M. Spiliopoulou, "C-denstream: Using domain knowledge on a data stream," in *Proc. of the 12th International Conference on Discovery Science*, ser. DS '09. Berlin, Heidelberg: Springer-Verlag, 2009, pp. 287–301. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-04747-3_23

[56] L. Li-xiong, K. Jing, G. Yun-fei, and H. Hai, "A three-step clustering algorithm over an evolving data stream," in *Intelligent Computing and Intelligent Systems, 2009. ICIS 2009. IEEE International Conference on*, vol. 1, nov. 2009, pp. 160 –164.

[57] "Kosmix," 2007, `http://www.kosmix.com/`.

[58] "Justspotted," 2010, `http://www.justspotted.com/`.

[59] "Trendistic," 2007, `http://trendistic.com/`.

[60] "Tweetmeme," 2011, `http://tweetmeme.com/`.

[61] "Tweetfeel," `http://www.tweetfeel.com/`.

[62] M. Mathioudakis and N. Koudas, "Twittermonitor: trend detection over the twitter stream," in *Proc. of the 2010 international conference on Management of data*, ser. SIGMOD '10. New York, NY, USA: ACM, 2010, pp. 1155–1158. [Online]. Available: http://doi.acm.org/10.1145/1807167.1807306

[63] T. Sakaki, M. Okazaki, and Y. Matsuo, "Earthquake shakes twitter users: real-time event detection by social sensors," in *Proc. of the 19th international conference on World wide web*, ser. WWW '10. New York, NY, USA: ACM, 2010, pp. 851–860. [Online]. Available: http://doi.acm.org/10.1145/1772690.1772777

[64] M. Cataldi, L. Di Caro, and C. Schifanella, "Emerging topic detection on twitter based on temporal and social terms evaluation," in *Proc. of the Tenth International Workshop on Multimedia Data Mining*, ser. MDMKDD '10. New York, NY, USA: ACM, 2010, pp. 4:1–4:10. [Online]. Available: http://doi.acm.org/10.1145/1814245.1814249

[65] S. Petrović, M. Osborne, and V. Lavrenko, "Streaming first story detection with application to twitter," in *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, ser. HLT '10. Stroudsburg, PA, USA: Association for Computational Linguistics, 2010, pp. 181–189. [Online]. Available: http://portal.acm.org/citation.cfm?id=1857999.1858020

[66] K. S. Charu C. Aggarwal, "Event Detection in Social Streams," in *Proc. of SDM Conference*, 2012.

[67] Q. Zhao, P. Mitra, and B. Chen, "Temporal and information flow based event detection from social text streams," in *Proc. of the 22nd national conference on Artificial intelligence - Volume 2*. AAAI Press, 2007, pp. 1501–1506. [Online]. Available: http://portal.acm.org/citation.cfm?id=1619797.1619886

[68] H. Sayyadi, M. Hurst, and A. Maykov, "Event Detection and Tracking in Social Streams," in *Proc. of International Conference on Weblogs and Social Media (ICWSM)*, 2009.

[69] J. Allan, V. Lavrenko, D. Malin, and R. Swan, "Detections, bounds, and timelines: Umass and tdt-3," in *In Proc. of Topic Detection and Tracking Workshop (TDT-3)*, 2000.

[70] L. Chen and A. Roy, "Event detection from flickr data through wavelet-based spatial analysis," in *Proc. of the 18th ACM CIKM*. NY, USA: ACM, 2009, pp. 523–532. [Online]. Available: http://doi.acm.org/10.1145/1645953.1646021

[71] H. Becker, M. Naaman, and L. Gravano, "Learning similarity metrics for event identification in social media," in *Proc. of the third ACM international conference on Web search and data mining*, ser. WSDM '10. New York, NY, USA: ACM, 2010, pp. 291–300. [Online]. Available: http://doi.acm.org/10.1145/1718487.1718524

[72] ——, "Beyond trending topics: Real-world event identification on twitter," in *Fifth International AAAI Conference on Weblogs and Social Media*, 2011.

[73] D. Ramage, S. Dumais, and D. Liebling, "Characterizing microblogs with topic models," in *ICWSM*, 2010. [Online]. Available: http://www.stanford.edu/~dramage/papers/twitter-icwsm10.pdf

[74] S. Papadopoulos, Y. Kompatsiaris, A. Vakali, and P. Spyridonos, "Community detection in social media," *Data Min. Knowl. Discov.*, vol. 24, no. 3, pp. 515–554, May 2012. [Online]. Available: http://dx.doi.org/10.1007/s10618-011-0224-z

[75] E. Bakshy, I. Rosenn, C. Marlow, and L. Adamic, "The role of social networks in information diffusion," in *Proceedings of the 21st international conference on World Wide Web*, ser. WWW '12. New York, NY, USA: ACM, 2012, pp. 519–528. [Online]. Available: http://doi.acm.org/10.1145/2187836.2187907

[76] L. Devroye, "Sample-based non-uniform random variate generation," in *Proc. of the 18th conference on Winter simulation*, ser. WSC '86. New York, NY, USA: ACM, 1986, pp. 260–265. [Online]. Available: http://doi.acm.org/10.1145/318242.318443

[77] H. Hotelling, "The Generalization of Student's Ratio," pp. 360–378, Aug. 1931.

[78] *Modern Multidimensional Scaling: Theory and Applications (Springer Series in Statistics)*, 2nd ed. Springer, Aug. 2005. [Online]. Available: http://www.worldcat.org/isbn/0387251502