Prédiction de l'activité dans les réseaux sociaux

François Kawala

UNIVERSITÉ DE GRENOBLE

THÈSE

Pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ DE GRENOBLE

Spécialité : **Informatique**

Arrêté ministérial :

Présentée par

François Kawala

Thèse dirigée par **Eric Gaussier**
et codirigée par **Ahlame Douzal**

préparée au sein **Laboratoire d'informatique de Grenoble**
et de **Ecole Doctorale Mathématiques, Sciences et Technologies de l'Information, Informatique**

Prédiction de l'activité dans les réseaux sociaux

Thèse soutenue publiquement le **8 octobre 2015**,
devant le jury composé de :

Guillaume
Jean-loup, Rapporteur
Linares
Georges, Rapporteur
Largeron
Christine, Présidente
Gaussier
Eric, Directeur de thèse
Douzal
Ahlame, Co-Directrice de thèse

Contents

List of Figures

List of Tables

Abstract

This dissertation is devoted to a social-media-mining problem named the activity-prediction problem. In this problem one aims to predict the number of user-generated-contents that will be created about a topic in the near future. The user-generated-contents that belong to a topic are not necessary related to each other.

In order to study the activity-prediction problem without referring directly to a particular social-media, a generic framework is proposed. This generic framework allows to describe various social-media in a unified way. With this generic framework the activity-prediction problem is defined independently of an actual social-media. Three examples are provided to illustrate how this generic framework describes social-media. Three definitions of the activity-prediction problem are proposed. Firstly the magnitude prediction problem defines the activity-prediction as a regression problem. With this definition one aims to predict the exact activity of a topic. Secondly, the buzz classification problem defines the activity-prediction as a binary classification problem. With this definition one aims to predict if a topic will have an activity burst of a predefined amplitude. Thirdly the rank prediction problem defines the activity-prediction as a learning-to-rank problem. With this definition one aims to rank the topics accordingly to theirs future activity-levels. These three definitions of the activity prediction problem are tackled with state-of-the-art machine learning approaches applied to generic features. Indeed, these features are defined with the help of the generic framework. Therefore these features are easily adaptable to various social-media. There are two types of features. Firstly the features which describe a single topic. Secondly the features which describe the interplay between two topics.

Our ability to predict the activity is tested against an industrial-size multilingual dataset. The data has been collected during 51 weeks. Two sources of data were used: Twitter and a bulletin-board-system. The collected data contains three languages: English, French and German. More than five hundred millions user-generated-contents were captured. Most of these user-generated-contents are related to computer hardware, video games, and mobile telephony. The data collection necessitated the implementation of a daily routine. The data was prepared so that commercial-contents and technical failure are not sources of noise. A cross-validation method that takes into account the time of

observations is used. In addition an unsupervised method to extract buzz candidates is proposed. Indeed the training-sets are very ill-balanced for the buzz classification problem, and it is necessary to preselect buzz candidates. The activity-prediction problems are studied within two different experimental settings. The first experimental setting includes data from Twitter and the bulletin-board-system, on a long time-scale, and with three different languages. The second experimental setting is dedicated specifically to Twitter. This second experiment aims to increase the reproducibility of experiments as much as possible. Hence, this experimental setting includes user-generated-contents collected with respect to a list of unambiguous English terms. In addition the observation are restricted to ten consecutive weeks. Hence the risk of unannounced change in the public API of Twitter is minimized.

Résumé

Cette étude est dédiée à un problème d'exploration de données dans les médias sociaux: la prédiction d'activité. Dans ce problème nous essayons de prédire l'activité associée à une thématique pour un horizon temporel restreint. Dans ce problème des contenus générés par différents utilisateurs, n'ayant pas de lien entre eux, contribuent à l'activité d'une même thématique.

Afin de pouvoir définir et étudier la prédiction d'activité sans référence explicite à un réseau social existant, nous définissons un cadre d'analyse générique qui permet de décrire de nombreux médias sociaux. Trois définitions de la prédiction d'activité sont proposées. Premièrement la prédiction de la magnitude d'activité, un problème de régression qui vise à prédire l'activité exacte d'une thématique. Secondement, la prédiction de Buzz, un problème de classification binaire qui vise à prédire quelles thématiques subiront une augmentation soudaine d'activité. Enfin la prédiction du rang d'activité, un problème de learning-to-rank qui vise à prédire l'importance relative de chacune des thématiques. Ces trois problèmes sont étudiés avec les méthodes de l'état de l'art en apprentissage automatique. Les descripteurs proposés pour ces études sont définis en utilisant le cadre d'analyse générique. Ainsi il est facile d'adapter ces descripteurs à différent média sociaux.

Notre capacité à prédire l'activité des thématiques est testée à l'aide d'un ensemble de données multilingue: Français, Anglais et Allemand. Les données ont été collecté durant 51 semaines sur Twitter et un forum de discussion. Plus de 500 millions de contenus générés par les utilisateurs ont été capturé. Une méthode de validation croisée est proposée afin de ne pas introduire de biais expérimental lié au temps. De plus, une méthode d'extraction non-supervisée des candidats au buzz est proposée. En effet, les changements abrupts de popularité sont rares et l'ensemble d'entraînement est très déséquilibré. Les problèmes de prédiction de l'activité sont étudiés dans deux configurations expérimentales différentes. La première configuration expérimentale porte sur l'ensemble des données collectées dans les deux médias sociaux, et sur les trois langues observées. La seconde configuration expérimentale porte exclusivement sur Twitter. Cette seconde configuration expérimentale vise à améliorer la reproductibilité de nos expériences. Pour ce faire, nous nous concentrons sur un sous-ensemble des thématiques non ambiguës en Anglais.

En outre, nous limitons la durée des observations à dix semaines consécutives afin de limiter les risques de changement structurel dans les données observées.

Remerciements

Cette thèse était une aventure, nombreux sont ceux qui y prirent part, sans eux cette aventure n'eût pas été possible. Écrire ces remerciements est l'occasion de décréter l'aventure terminée et d'en présenter les acteurs. Il est évident que certains seront oubliés, à tous ces oubliés, mes remerciements, bien qu'impersonnels, sont véritablement sincères. Il me tient donc à coeur de remercier:

Les membres du jury: Christine Largeron, Jean-loup Guillaume et Georges Linares. Tous trois ont témoigné d'un réel intérêt pour les travaux présentés. Leurs remarques et conseils avisés m'ont permis d'améliorer significativement ce manuscrit. Merci d'avoir contribuer à ce travail.

Mes encadrants: mon directeur de thèse Eric, ma co-Directrice Ahlame, et mon Responsable industriel Eustache. Ils ont tous trois fait preuve d'une patience sans limite face à mes hésitations, et mes incessantes remises en question. Leurs conseils, en tout domaines furent précieux, du choix d'un algorithme jusqu'aux sentiers de VTT.

Mes collègues Grenoblois et Parisiens. Gilles, Julien, Brice, vous avez écouté debout ou assis, trois années de tâtonnement vers mon problème du moment, vos commentaires et vos conseils ont été précieux. Tristan, Olivier et Aurélien, de l'aventure vous n'avez connu que le dernier chapitre, merci de m'avoir aidé à l'écrire.

Ma famille, et plus particulièrement mes parents Michèle et Bernard, pour leur soutien inconditionnel et pour m'avoir toujours tout donné. Vous m'avez permis d'être ce que je suis aujourd'hui. Emmanuel et Anne pour m'avoir montré tant d'astuces quand j'étais petit. Jean-luc et Chantal, pour avoir laissé le robot au placard, et surtout pour tout le reste.

Mes amis comme: Mymy pour ces milliers de kilomètres, avec des chrono toujours plus beaux mais surtout en discutant de tout. Maniök, pour toujours savoir quand il est temps de tirer mon doigt et ton point de vue unique à bien des égards. Mégah pour le 56 Avenue Jeanne d'Arc, pour les rushs corniche, les mass goules, et les soirées pizza. Tsahal, pour avoir fait de nos anniversaires des moments uniques, pour ton oreille douce, et tes ronflements sourds. Dorette, pour me rappeler qu'il y a toujours un peu plus bo-bo que moi. Tucki, pour le 4 place Saint Bruno, le home cinéma, les stouts, et toutes ces occasions de refaire le monde. Noursåå, pour les "basket-piscine", pour ton doigt de pied, pour toutes ces belles soirées passées et à venir. Céline, de nous donner envie de voyager et d'apprendre. Fafa de t'occuper de Tucki et de la planète. Dédé d'être la preuve vivante que l'on peut être Ardéchois sans pour autant aimer le fromage de chèvre. Lucette, d'avoir toujours tout donné pour gagner au Trivial Pursuit®. Bourg, pour l'usage outrancier du passé simple, l'introduction de ces remerciements, et m'avoir guidé lors de mon adolescence. Koko, pour m'avoir conforté dans mes pires travers bobo-geek mais surtout gaucho-geek. Grincheux, pour tes bons petits plats, ton amour du bon vin et des beaux mots. Joe pour tes bons plans, et tes conseils affûtés de vieux loup de laboratoire. Yoyo pour m'avoir fait découvrir le tsipouro à l'improviste et sans prévenir, pour les melomakarona, la féta, tous tes conseils et tes invitations au voyage.

Enfin, le témoin privilégié, si l'on ose dire, de cette aventure: Aurélia. Elle en a supporté les conséquences chaque jour sans faillir. Pour moi, il n'est personne qui sache si bien qu'elle partager les joies et adoucir les peines. Sans elle, point d'aventure, mes remerciements ne seront jamais assez grands.

Introduction

Internet grew dramatically in the past decade and is about to become pervasive. There are now 2.9 Billions Internet users, whereas they were only 900 Millions in 2004. A direct consequence of this growth is that Internet became widely used to daily human interactions. At the same time, the amount of data that flows through Internet grows quickly. A significant part of these data are automatically recorded and stored in expectation of their future value. This new gold-rush is motivated by the expected capacity of algorithms to extract actionable patterns from the collected data, and theirs subsequent incomes. This is particularly true for the social-media. As presented in Section 1.1, the term social-media is an umbrella term that designates Internet-based services that allows to exchange content such as: pictures, movies, songs, writings. These contents are frequently referred to as user-generated-contents. The data that is expected to have financial worth is divided in three categories. Firstly the user-generated-contents themselves (*eg.* messages, pictures). Secondly, the user-interactions (*eg.* who is friend with whom; who shared who's content). Thirdly, the implicit user-activity (*eg.* as query logs in a search engine).

Data mining is defined as the extraction of actionable patterns from data. As such, data mining is the research area that would produce the expected value out of social-media data. Data mining is made possible at a large scale by statistical methods that allow automated extraction of decision rules from data. The research effort dedicated to these statistical methods is known as machine learning. The data collected in social-media have specificities. Firstly, most of the data is unstructured. Secondly the user-generated-contents follow few stylistic constraints. Thirdly, specific conventions are used in some user-communities. Lastly the quantity of data to be processed is tremendous and therefore requires to use highly scalable approaches. Due to these specific challenges, data mining applied to social-media might be referred specifically as *social-media-mining*. This dissertation is devoted to a *social-media-mining* problem named the activity-prediction

problem. Broadly, in the activity prediction problem one aims to predict the number of user-generated-contents that will be created about a topic in the near future. Section 1.2 presents *social-media-mining* from a practitioner point of view. Firstly, Section 1.2.1 presents the characteristics of graphs of users that are defined in social-media. Secondly, Sections 1.2.2 and 1.2.3 presents two problems related to activity-prediction: the community detection and the information diffusion problems. The collection of data from social-media is discussed in Section 1.2.4. Afterwards, key concepts of machine learning are presented in Section 1.2.5.

Activity-prediction is an open question and draws an important amount of work, as presented in Section 1.3. Firstly, Section 1.3.1 takes an inventory of concepts that are the most often used in these studies. Afterwards three approaches for the activity-prediction are distinguished. The taxonomies of the spikes of collective attention are presented in Section 1.3.2. These studies aim to understand and classify the spikes of collective attention. A spike of collective attention is observed when a topic (*eg.* Christmas) becomes one of the top most popular topic in a whole social-media. The spikes of collective attention affect few (*eg.* dozens) topics at a time, are highly dynamic, and mostly news related. Secondly, Section 1.3.3 presents the studies dedicated to predict the activity of a topic. In order to measure the activity of a topic one assigns each user-generated-content to one topic. Then he-or-she uses a activity measure that takes into account all the user-generated-contents that were assigned to a topic. Hence, the user-generated-contents that are considered for a topic are not necessary related to each other. On the contrary, Section 1.3.4 presents user-generated-content activity-prediction. The task is then to predict the quantity of reactions that a particular user-generated-content will trigger on a specific social-media. In this case all the user-generated-contents that are considered are explicitly related.

The three activity-prediction problems studied in this dissertation aim to predict the activity of the topics. In order to study these problems without referring directly to a particular social-media, a generic framework is proposed in Section 2.1. This generic framework allows to describe various social-media in a unified way. With this generic framework the activity-prediction problems are defined independently of an actual social-media. Three examples are provided to illustrate how this generic framework describes social-media. The first example, presented in Section 2.1.2, illustrates how Twitter is described with this generic framework. The second example, presented in Section 2.1.3, illustrates

how Facebook is described with this generic framework. The last example, presented in Section 2.1.4, illustrates how a bulletin-board-system is described with this generic framework. The activity-prediction problems are formally presented in Section 2.2. These three problem address distinct applicative needs. Firstly the magnitude prediction problem, presented in Section 2.2.1, formalizes the activity-prediction as a regression problem. With this formalization one aims to predict the exact activity of a topic. Secondly, the buzz classification problem, presented in Section 2.2.2, formalizes the activity-prediction as a binary classification problem. With this formalization one aims to predict if a topic will have an activity burst of a predefined amplitude. Thirdly the rank prediction problem, presented in Section 2.2.3, formalizes the activity-prediction as a learning-to-rank problem. With this formalization one aims to rank the topics accordingly to theirs future activity-levels. These three problems are tackled with state-of-the-art machine learning approaches. Section 2.3 presents the features used to tackle the three activity-prediction problems. These features are defined thanks to the generic framework. Therefore these features are easily adaptable to various social-media. There are two types of features. Firstly the features which describe a single topic, as presented in Section 2.3.1. Secondly the features which describe the interplay between two topics, as presented in Section 2.3.2.

Our ability to predict the activity within each of the three activity-prediction problem is tested against an industrial-size multilingual dataset. The data was collected during 51 weeks on Twitter and a bulletin-board-system run by the company Purch. This company funded and supported this study. Purch granted us access to privileged data. The collected data contains three languages: English, French and German. More than five hundred millions user-generated-contents were captured during the data collection. Most of these user-generated-contents are related to computer hardware, video games, and mobile telephony, as described in Section 3.1. The collection of this amount of data necessitated the conception of a daily routine. In addition the data was prepared so that commercial-contents and technical failure are not sources of noise. A cross-validation method that takes into account the time of observations is defined in Section 3.2. Indeed, the activity-prediction problems are time dependent and classical cross-validation methods might induce experimental noise. In addition an unsupervised method to extract buzz candidates is proposed. Indeed the training-set is very ill-balanced for the buzz classification problem, and it is necessary to preselect buzz candidates. The activity-prediction problems are studied within two different experimental settings. The first experimental setting includes data from Twitter and the bulletin-board-system, on a

long time-scale, and with three different languages. The second experimental setting is dedicated specifically to Twitter. This second experiment aims to increase the reproducibility of experiments as much as possible. Hence, this experimental setting includes user-generated-contents collected with respect to a list of unambiguous English terms. In addition the observation are restricted to ten consecutive weeks. Hence the risk of unannounced change in the public API of Twitter is minimized. Section 3.3 presents the experiments related to the buzz classification problem for each of these experimental settings. Sections 3.4 and 3.5 present the counterparts experiments for the magnitude prediction and the rank prediction, respectively.

Chapter 1

State of the art

Contents

Introduction

This study fits into the research area of online-social-media mining. This research area, also named *social-media-mining*, is a fast growing interdisciplinary domain dedicated to the study of human interactions on the Internet. The definition of social-media has not

yet reach a general consensus, thus this chapter reviews two definition attempts. Without being exhaustive, this chapter gives a general overview of challenges that are addressed by computer scientists in the social-media mining area.

For the last decade the general public have widely adopted the Internet. In 2004 the United Nations counted 900 Millions Internet users, there are now 2.9 Billions Internet users. One consequence of this broad adoption, is the growth of human communication mediated by Internet. Surveys confirm the widespread usage of Internet for this communicational purposes. For instance, the Pew research institute, a private U.S. research group, estimates that 73% of the U.S. adults use at least one social media [49]. Moreover, quantcast, a company that ranks websites according to their traffic, estimates that social media sites such as: Facebook[1], Twitter[2], and Linkedin[3] are all among the fifteen most visited web site, for June 2014, in the U.S. [133].

A significant part of these computer mediated communication are automatically recorded. These records of computer mediated communication constitute an unprecedented source of human interactions to be studied. The study of these records are tackled through social sciences and computer science. These records are threefold, they contain: (a) the content created by the users (*eg.* messages, photos); (b) the traces of theirs interaction (*eg.* who is friend with whom; who shared who's content); and (c) implicit traces of user-activity (*eg.* as query logs in a search engine) which are generally not made available by their owner. The content created by the users is commonly referred to as user-generated-content. The best known definition of the user-generated-content is the one from Vickery and Wunsch-Vincent [154] proposed on the behalf of the Organisation for Economic Cooperation and Development. As per Vickery and Wunsch-Vincent, a user-generated-content: is (a) "*made publicly available over the Internet*"; (b) is "*created outside of professional routines and practices*"; (c) shows a "*certain amount of creative effort*". A user-generated-content can include text, videos, images, songs, and so on. For instance, a photo published on a social-media or a note in a weblog are user-generated-contents. The user-generated-contents are unstructured, they have few stylistic constraints as they are created outside professional practices, and they are produced at a fast-paced rhythm [128, 122].

[1]Ressource available at `http://www.facebook.com`
[2]Ressource available at `http://www.twitter.com`
[3]Ressource available at `http://www.linkedin.com`

The *social-media-mining* research area can be thought of as a special form of data mining. As such the *social-media-mining* would be defined as the extraction of useful, and novel, knowledge from the social-media data. In regard to the aforementioned social-media characteristics, the extraction of useful and novel knowledge from social-media should be: (a) efficient and scalable in order to cope with the unprecedented, yet growing, amount of user-generated-contents; (b) resilient to the unstructured and noisy nature of user-generated-contents; (c) able to combine user-generated-contents and the traces of interactions between users (*eg.* who is friend with whom). This chapter outlines *social-media-mining* from a practitioner point of view, and is structured as follows. The first section presents several definitions of a social-media and describes the recent social-media evolutions. The second section of this chapter presents the networks of users observed in social-media, and how to apply machine learning to social-media. The third section of this chapter present the research areas devoted to the popularity prediction problem in social-media.

1.1 Definition of a social-media

As per Boyd, "social-media" is an umbrella term, synonym of computer-mediated communication, that designates the *"tools, services, and applications that allow people to interact with others using network technologies"* [30]. As such, social-media ranges from short messages scripted to social-network-site, and covers indistinctly private, semi-public, and public communications. Nonetheless, up to now, *social-media-mining* is devoted to Web-based social-media. Those Web-based social-media are applications that pertain to Web 2.0. The Web 2.0, a term first coined in 1999, is defined by Easley and Kleinberg [47] as a set of technologies that allows to:

- collectively create and maintain shared content;
- shift personal data from personal computer to online services (*eg.* videos, photos);
- link peoples in the same way that documents are linked in the web.

As an illustration of the Web 2.0, one can consider a social-media such as Flickr[4]. Indeed, this social-media allows people to share photographies and to comment them. As such,

[4]Ressource available at `http://www.flickr.com`

Flickr shifts contents from one's personal computer to the Internet, and allows to share that contents. Numerous web-based social-media have redundant purposes. Therefore, Kaplan proposes to classify social-media into six groups defined with respect to their purpose [85]. Four of those groups describe web-based social-media, they are presented below.

- The collaborative projects, that allow to create and maintain contents collaboratively. In this case users are not necessarily identified. The best-known collaborative project is Wikipedia[5], a collaborative encyclopedia. The "question & answer" platforms such as Yahoo Answers[6], and the forums such as StackOverflow[7] are collaborative projects where the content creation is driven by the need of a solution to a specific problem.

- The blogs and microblogs, that allow to create and share contents. Contrary to the collaborative projects, the authorship is promoted in blog and microblogs. Indeed, each author is explicitly associated to a time ordered list of user-generated-contents. The contents shared through blog and microblogs vary from one author to other. For instance, a blog might be used as personal diary, or as a technical notepad.

- The content communities, that allow to share one specific content type including: videos, images, songs, or slides. For instance, Flickr is a content community that is dedicated to photographies. On the contrary YouTube or Vimeo[8] are content communities dedicated to videos. Kaplan, notes that the these communities aren't focused on users. Instead, these communities are focused on the contents: "*Users [...] are not required to create a personal profile page; if they do, these pages usually only contain basic information*".

- The social-networking-sites, that allow users to "*connect [to each others] by creating personal information profiles, inviting friends and colleagues to have access to those profiles*". This description is completed per Ellison and Boyd [29]. They define the early stage social-network-site as web-based applications that "*allow individuals to (1) construct a public or semi-public profile within a bounded system, (2) articulate a list of other users with whom they share a connection, and (3) view and traverse their list of connections and those made by others within the system.*".

[5]Ressource available at `http://www.wikipedia.org`
[6]Ressource available at `http://answers.yahoo.com`
[7]Ressource available at `http://www.stackoverflow.com`
[8]Ressource available at `http://www.vimeo.com`

Kaplan compares these four social-media classes with respect to their media richness and their self-disclosure level. The media richness refers to the possible complexity of the interaction. For instance, text based interaction (*eg.* as in blogs) is considered as simpler than multimedia based interaction (*ie.* text combined with photo, video, and so on). The self-disclosure level, refers to the ownership level on the contents that one shares. This comparison, is presented in Table 1.1.

		Media richness	
		Low	Medium
Self-disclosure	High	blogs	Social networking sites
	Low	Collaborative projects	Content communities

Table 1.1: Classification of Web-based social-media as proposed by Kaplan. The original classification has two additional type of social-media. These two social-media types are based on video-games, thus they are not represented in this Table.

The classification proposed by Kaplan provides a snapshot of the social-media history. The social-media continuously evolve since then. As an example of this evolution consider how the broad adoption of smartphones influenced social-media. The smartphones that locates users through GPS enabled the creation of dedicated social-media. In addition social-media that existed before the generalization of smartphones started to leverage GPS location to enrich their users experience. An example of a social-media dedicated to GPS usage is Foursquare[9] which allows one to locate his-or-her nearby friends. On the contrary, Facebook is an example of social-media that was not originally devoted to use GPS, but which evolved in order to use GPS coordinates. Social-media are continuously evolving in order to fulfill the users-expectations, and to steer them to new usages. A consequence of this evolution is that today the blogs, the microblogs, the content communities, and the social-network-siteare no longer clearly distinguishable. More precisely:

- The social-network-sites shifted progressively to a network of user-generated-content streams as noted Ellison and Boyd [52]. As such social-network-sites became increasingly content-centric. Thus social-network-sites became closer to the content communities.

[9]Ressource available at `http://www.foursquare.com`

- The content communities, which had a poor user representation, now have a richer user representation. For instance, their users can now subscribe to each others, have favorites contents and so one. Thus content communities tend to social-network-sites.

- The Microblog platform, such as Twitter, eased the sharing of richer media such as photo or video. Such an evolution bring microblog closer to content communities. Another change is that some blog platform, such as Tumblr[10], shifted towards a network of user-generated-content streams and thus are now closer to social-network-sites with additional features dedicated to the production of textual content, as for instance source code formatting.

As a consequence of this evolution, Ellison and Boyd defined the social-network-site in a unified way. Their definition fits all the web-based social-media exception made of the collaborative projects. Therefore, in the remainder of this dissertation, the social-media are considered as web-based services, that are either collaborative projects (*eg.* Wikipedia, StackOverflow), or services that fits in the definition proposed per Ellison and Boyd. This definition is as follows: A social-network-site is system such that: (a) "*users have uniquely identifiable profiles that consist of user-supplied content provided by other users, and/or system-provided data*"; (b) "*users can consume, produce, and/or interact with streams of user-generated content provided by their connections on the site*"; (c) "*users can publicly articulate connections that can be viewed and traversed by others*" [52].

1.2 General considerations

As per Zafarani *et al.*, the *social-media-mining* mining is the "*process of representing, analyzing, and extracting actionable patterns from social media data*" [167]. As defined in Section 1.1, a social-media might be a collaborative project, or consists of users that are organized in a user-network and exchange user-generated-contents. This section is divided in two parts. The first part provides general information about the user-networks observed in social-media. The second part presents machine learning and how it applies in *social-media-mining*. This section, by no means intends to outline the theoretical bases of the field related to *social-media-mining*, as for instance : machine learning, graph mining, data mining or natural language processing. Indeed, numerous reference work

[10]Ressource available at `http://www.tumblr.com`

provide a comprehensive outline of the aforementioned theoretical bases. Therefore the readers interested in:

- Graph-mining, are referred to the work of Aggarwal [10, 11], Cook and Holder [42], Wasserman [156] or Nettleton [120] for a work devoted to social networks;
- Machine learning, are referred to the work of Barber [18], Bishop [24], Mohri [115], Murphy [117] for a work devoted to generative approaches, or Hastie *et al.* [77];
- Data mining, are referred to the work of Witten and Frank [159] for a general purpose introduction, Jurafsky [83] or Mannning and Schütze [110] for natural language processing questions. Note that in the social-media specific methods are necessary as outlined by Han and Baldwin [76].

1.2.1 Characteristics of the user-network in social-media

The graph framework is used since the 16^{th} century with the famous Euler's problem of the "Seven Bridges of Königsberg". The research on graphs led to the formalization of various problems such as: the graph coloring problem, the route problems, the covering problems, the network flow problems, or the subgraph matching problem. These problems are formally defined in several introductory books [27, 150, 158], therefore they aren't presented is this section. However the graph-mining is closely related to *social-media-mining*, thus key concepts of graph theory are presented in this section. Afterwards, a general picture of the users network observed in social-media is outlined. Finally, practical problems such as the collection of the user network from a social-media are discussed.

Key concepts of the graph theory

A graph is convenient framework to study problems from various domains, including, but not limited to, social science, biology, and computer science. Knuth defines an undirected graph as "*a set of points (called vertices) together with a set of lines (call edges) joining certain pairs of distinct vertices. There is at most one edge joining any pair of vertices*" [90]. Two vertices are called "adjacent" when it exists one edge between them. A sequence of vertices $[v_1, \ldots, v_{n-1}, v_n] \in V^n$ is called a path when each pair (v_i, v_{i+1}) is adjacent. Consider $u, v \in V^2$, a path such that $v_1 = u$ and $v_n = v$ is a path of length n that connects u and v. It might exists several distinct paths between u and v, their lengths may vary.

There are two additional types of graph: the weighted graph, and the directed graph. In a weighted graph, every edge is associated to a weight in $\mathbb{R}$. In a directed graph, every edge has a direction, therefore a vertex u is adjacent to a vertex v if it exist an edge that goes from u to v. Contrary to the undirected graph, in the directed graph there are a most two edges joining any pair of vertices. Figure 1.1a presents a directed graph with three vertices and four edges. A graph can be described through the adjacency of its

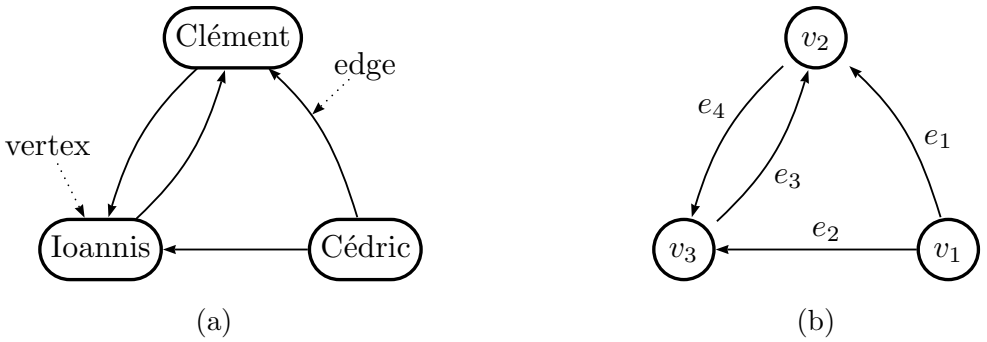

Figure 1.1: This Figure, illustrates, on its left part, a user-network with three users (*ie.* vertices): Clément, Ioannis and Cédric. This user-network has both symmetric and asymmetric relations (*ie.* edges). The relation of Clément and Ioannis is symmetric, but the relation of Ioannis and Cédric is asymmetric. The right part of this Figure shows the same graph with additional names in order to illustrate the Adjacency and Incidence matrices.

vertices, or through the association of its edges to its vertices. These two representations can be encoded in matrices or lists. Consider for instance a directed graph named $\mathcal{G}$, which edges set is E, and vertices set is V with $|V| = n$. Usually, n is referred to as the order of $\mathcal{G}$. The adjacency matrix of such a graph, named $\mathbf{A}$, is of dimension $n \times n$. The cell $\mathbf{A}_{i,j}$ is 1 if it exists an edge between the ith and jth vertices. The adjacency matrix of the graph $\mathcal{G} = (V, E)$, is defined as follows:

$$\mathbf{A}(\mathcal{G})_{i,j} = \begin{cases} 1 & \text{if } \ (i,j) \in E \\ 0 & \text{else.} \end{cases}$$

In the case of an undirected graph, the matrix $\mathbf{A}$ is symmetric. The incidence matrix of $\mathcal{G}$, indicates which edges starts (*resp.* ends) from (*resp.* at) a vertex. Consider that the set of edges contains m elements such as $E = \{e_1, \ldots, e_m\}$. Usually, m is referred to as the size of $\mathcal{G}$. In this case, the incidence matrix, named $\mathbf{C}$, is of dimension $n \times m$. The value of $\mathbf{C}_{i,j}$ is 1 if the edge e_j starts from vertex i, -1 if the edge e_j ends at vertex i,

and 0 otherwise. The incidence matrix is defined as follows:

$$\mathbf{C}(\mathcal{G})_{i,j} = \begin{cases} 1 & \text{if } \exists k \in V \text{ s.t. } e_j = (i,k) \\ -1 & \text{if } \exists k \in V \text{ s.t. } e_j = (k,i) \\ 0 & \text{else.} \end{cases}$$

The degree of a vertex $v_i \in V$ is the number of edges that are incident to it. The degree is equal to the sum of the i^{th} row of the adjacency matrix, and defined as follow: $\texttt{degree}(v_i) = \sum_{j=1}^{n} \mathbf{A}(\mathcal{G})_{i,j}$. The in-degree of a vertex is defined only in a directed graph. The in-degree of the vertex of $v_i \in V$ is the number of edges that ends at v_i, it is defined as follows:

$$\texttt{in\_degree}(v_i) = \sum_{j=1}^{n} \mathbb{1}\left[\mathbf{C}(\mathcal{G})_{i,j} < 0\right]$$

Where $\mathbb{1}$ is the indicator function. The out-degree of a vertex is defined only in a directed graph. The out-degree of the vertex of $v_i \in V$ is the number of edges that starts from v_i, it is defined as follows:

$$\texttt{out\_degree}(v_i) = \sum_{j=1}^{n} \mathbb{1}\left[\mathbf{C}(\mathcal{G})_{i,j} > 0\right]$$

As an illustration, consider the graph presented in Figure 1.1b, the degree values for the vertex v_1 are as follows: $\texttt{degree}(v_1) = 2$, $\texttt{in\_degree}(v_1) = 0$ and $\texttt{out\_degree}(v_1) = 2$. The adjacency and incidence matrices for the graph presented in Figure 1.1b are:

$$\mathbf{A}(\mathcal{G}) = \begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \end{array} \begin{array}{c} \begin{array}{ccc} v_1 & v_2 & v_3 \end{array} \\ \begin{bmatrix} 0 & 1 & 1 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \end{array} \qquad \mathbf{C}(\mathcal{G}) = \begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \end{array} \begin{array}{c} \begin{array}{cccc} e_1 & e_2 & e_3 & e_4 \end{array} \\ \begin{bmatrix} 1 & 1 & 0 & 0 \\ -1 & 0 & -1 & 1 \\ 0 & -1 & 1 & -1 \end{bmatrix} \end{array}$$

The adjacency and incidence matrices might be significantly sparse, therefore one should consider an appropriate memory representation for these matrices. It exists other representations that are tailored for more specific questions, as for instance the Lapacian matrix. This matrix is equal the difference of the degree matrix and the adjacency matrix, where the degree matrix is a diagonal matrix defined as $\mathbf{D}_{i,i} = \texttt{degree}(v_i)$.

A direct application of the graph theory is the graph-mining. The graph-mining received a great deal of attention, indeed its has various application domains including, but not limited to: biology, chemistry and computer science. As per Cook and Holder, the graph-mining, or graph-based data mining, is "*the [automated] extraction of novel*

and useful knowledge from a graph representation of data." [42]. Graph-mining is a form a data-mining closely related to *social-media-mining*, indeed social-media users can be represented within a graph. In this user-network, the vertices represent users, and the edges represent relations among them. Figure 1.1a presents a toy example for such a graph, it has three users (*ie.* vertices): Clément, Cédric, and Ioannis. In "real world" user-networks symmetric and asymmetric relations are observed. For instance Facebook relies on a symmetric relation: the "friendship". On the contrary Twitter proposes an asymmetric relation. The section outlines the principal characteristics of the graphs observed in social-media and the Web. Afterward three questions related to Graph mining and social-media are presented. Firstly, how to extract user communities from the social-media? Secondly, how to model the information diffusion among users of a social-media? Thirdly, how to capture the graph of users from the social-media ?

Characteristics of the user-network in a social-media

Aggrawal [10] notes that, in order to propose efficient solutions to graph-mining problems, one has to consider the characteristics of the data to be processed. Indeed, the structure of a graph may vary with respect to the considered data. For instance, compare a graph that represents chemical compounds, with a graph that represent the Web. In the former, each vertex represents a chemical element (*eg.* Hydrogen, Oxygen, and so forth). There are few different chemical elements, therefore it is likely to observe numerous vertices with the same attributes. In such a case, to apply pattern mining is sensible [42, 10]. On the contrary, in the Web-graph, a vertex represents a unique resource (*eg.* a web page), and an edge represents a hyperlink between two of such resources. Therefore, in this situation, to apply pattern mining is not straightforward. More generally, there are three important research threads that are related to the specificities of the data to be processed: (a) The definition of efficient graph representation (*eg.* as the usage of sparse data structure, or compression methods). The efficiency is a key problem in graph-mining, indeed numerous problems are NP-Hard and requires the usage of an heuristic. (b) The control of the sampling bias. This question is non trivial, indeed to obtain the complete graph or even just a random sample is not always possible. In such a situation, it is hard to decide if a sample is representative of the whole user-network. This question is studied by Guillaume and Latapy [73] and da Fontoura Costa and Travieso [65]. (c) The definition of graphs generator. These generators are used to produce synthetic graph under specific characteristic. Goldenberg *et al.* propose a survey of these generative models for the

Web-graph [67].

The main measures for the characterization of a graph are based on its size, its degree distribution, and its paths. There are numerous studies on the Web-graph characteristics [54, 33, 58, 104, 144, 39]. Some insightful observations about the Web-graph are summarized below. However, the body of search devoted to complex network is beyond the scope of this dissertation. It is noteworthy that in this research area, a vertex is often referred to as "a node", and an edge is referred to as "a link".

- As per Broder, *et al.* [33], the Web graph is divided into: a strongly connected component (*ie.* a set of nodes where it exists a path between each pair of nodes in the set) and two other sets of nodes sets. The first set contains the nodes that are reachable from the strongly connected component. The second set contains the nodes from which the strongly connected component can be reached.

- As per Faloutsos *et al.* [54] the degree distribution of the Web graph follows a power law distribution. Clauset *et al.* [39] note that a better approximation is obtained with a power law with exponential cutoff. In a graph whose in degree distribution follows a power law, the probability for a node to have a degree equal to k is proportional to $k^{-\gamma}$ with $\gamma > 1$. This probability is $k^{-\gamma}e^{\beta k}$ for a graph that follows a power law with exponential cutoff.

- As per Leskovec *et al.* [104], the average path length decrease over time (*ie.* as the order of the graph increase), and the number of links increase faster than the number of nodes.

In the same way, the characteristics of the user-networks observed in social-media have been extensively studied [93, 13, 114, 65, 94]. It is noteworthy that some characteristics of the user-network observed in social-media were first devised in social science about social-graph (*ie.* a graph of human beings where links represent a form of relationship). For instance, Milgram [112] formulated the famous *small-world* phenomenon in the 1960s after having measured empirically the average path length between two U.S. peoples. The structure of the social-graph was also investigated, for instance Granovetter [72] argues that there are "weak" and "strong" links in a social-graph, such that strong links have an higher clustering coefficient. However, it is harder to draw general conclusions about the user-networks observed in the social-media. Indeed, each social-media has specific usage

patterns that may influence the user-network structure. For instance, in Twitter[11], until 2009, a user could not follow (*ie.* to be connected to) more than two thousand users. In addition in Twitter users are invited to follow at least 20 celebrities when they create their account. In 2010 the degree distribution was reflecting this limitation [94]. The studies of the user-networks characteristics for: Flickr, Twitter and LiveJournal, are now presented.

Kumar *et al.* [93] study Flickr and Yahoo! 360° a discontinued social-media. In Flickr, they observe that the majority of the nodes are either: disconnected (*ie.* their degeree is equal to zero), or belongs to a "giant component" that is strongly connected. Authors furthermore note that the nodes outside the "giant component" are organized in star shaped groups. Their definition of star shaped group is as follows: "*it has one or two nodes (centers) that have an edge to most of the other nodes in the component and it contains a relatively large number of nodes that have an edge solely to one of these centers*". The diameter of the giant component is also studied. The diameter is defined as maximum length for all the shortest paths. As the diameter is error-prone, authors use two related measures: the average diameter and the effective diameter. The former is defined as the average path length between two randomly chosen nodes. The effective diameter is the 90^{th} percentile of the shortest paths lengths distribution, this measure is defined in [104]. Authors report that giant component observed in Flickr has an average diameter of 6.01, and an effective diameter of 7.61. In addition, when one removes the nodes with a degree equal to 1 from the giant component, the average diameter decreases to 4.45. In Flickr, the ratio of symmetric relationship out of asymmetric ones, named "reciprocity level", has a value of 68%. Finally, Flickr is observed over time, the main conclusion is that the densification phenomenon, as described per Leskovec [104] for the web-graph, is also observed in Flickr.

Kwak *et al.* [94] found comparable results for the Twitter user-network. The in-degree distribution of the user-network follows a power law of exponent 2.276 with a cutoff at 10^5. The nodes with more than 10^5 in links (*ie.* followers) are more numerous than it would be predicted by the power-law. This situation might be related to the one observed in Cyworld[12]. Yong-Yeol *et al.* [13] studied Cyworld, and described a two scales in-degree distribution that is shaped as power-law, but has an heavy-tail. Authors posit that this situation is due to celebrities that readily communicate with their public. It is note-

[11] Section 2.1.2 illustrates the key mechanisms in Twitter, as for instance the "follow" relationship.

[12] Ressource available at `http://cyworld.co.kr`

worthy that in the case of Cyworld author had access to the whole user-network, a rare opportunity. Several abnormalities are observed in the Twitter in-degree distribution. Authors explain that these abnormalities are due to features such as the proposal, for a new user, of 20 popular account to follow. The reciprocity level in Twitter is significantly lower than in Flickr, with a value of 22%. Authors report that giant component observed in Twitter has a diameter of 18, but an effective diameter of 4.8. The link formation in the user-network is also investigated with respect to the geospatial distance between users. The distance between two users is estimated through time zones as it is the most reliable information. Authors concludes that, except for celebrities, a user tends to be connected to users from a single location.

Mislove *et al.* propose a study on four different social-media: Youtube, Flickr, a blogging service named Livejournal[13], and a discontinued social-network-site named Orkut. A particular attention is payed to the quality of the studied sample, and an estimation of the graph sample size is reported for each social-media. Authors note that, contrary to earlier work, their samples might be considered as representative, excepted for Youtube. Indeed, they were not able to estimate the number of Youtube users. In the four social-media, the reciprocity levels are consistent with the previously reported results. For instance, Flickr and LiveJournal have a reciprocity level of 62% and 73.5%, respectively. A power law distribution is fitted to node in-degree and out-degree distributions. The parameter of the power law is fitted by a maximum likelihood estimation. The results are then evaluated with the Kolmogorov-Smirnov goodness-of-fit metric, as proposed by Clauset *et al.* [39]. The power-law coefficients for Flickr are 1.74 and 1.78 for the in-degree and the out-degree, respectively. The power-law coefficients for Livejournal are 1.59 and 1.65. These values reveal a significant difference with the Web-graph. The diameters reported in this study are comparable to the previously reported results, for Flickr and Livejournal the average diameters are 5.67 and 5.88 respectively.

Besides these specific observations, Mislove *et al.* propose a general view of the characteristic of user-networks observed in social-media. This general view is summarized as follows. The node degree distribution follows a power-law. The average diameter is lower than 6, which is significantly lower than the value observed for the Web. Indeed, Broder *et al.* [33] report an average diameter of 16.12 for their observation of the web. The user-network has tightly linked users groups constituted of low-degree nodes. Inversely, the

[13]Ressource available at `http://www.livejournal.com`

high-degree nodes belong to several users groups. The clustering coefficient is therefore "*inversely proportional to the node degree*". The low average diameter is explained by the existence of an highly correlated core, which contains about 10% of the nodes, and connects the different users groups.

1.2.2 Community detection and social-media

The structure of the user-network observed in the social-media, is often considered to influence the spread of the information and its subsequent activity. Therefore, the research area devoted to the detection of communities in graph is now outlined. This research area is originated in graph clustering, and led to a tremendous research effort. Therefore, this section by no means intends to list exhaustively the works related to community detection. Instead this section presents the latest lines of inquiry for the community detection. For a broader discussion on this problem, the readers are referred: to Furtunato [57], Schaeffer [143], or Aggrawal [11]; to Xie *et al.* [161] for a survey dedicated to overlapping community detection; and to Malliaros and Vazirgiannis [109] for a survey dedicated to community detection in directed graphs. The first part of this section introduces the three definition of community. Afterward the current lines of inquiry are presented.

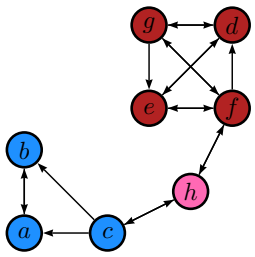

Figure 1.2: This Figure illustrates a user-network that might be divided in three communities which are illustrated with different colors.

The community detection problem is intuitive, indeed, the human brain is designed to clusters visual inputs seamlessly. For instance, the detection the communities is quite instinctive for the toy example presented in Figure 1.2. However, the definition of a community has three formulations: the disjoint community, the overlapping community, and the ego-community.

In the disjoint definition, a community is a densely connected subset of nodes. In this

subset, nodes are more likely to be connected to each others than to the remaining of the graph. This definition entails that a vertex belongs to one single community. In such a case, the community detection amounts to a partition of the set of nodes. This line of inquiry have received a great deal of attention, as described in the survey by Fortunato [57]. A substantial part of these work rely on the measures of betweenness and modularity. The betweenness estimates how often an edge is located between two communities. The modularity gives the quality of a partition by comparing it to a random graph. The seminal work for these measures is from Grivan and Newman [64]. Numerous variations based on these measures were proposed.

In the overlapping definition, a vertex might belong to several communities. This definition is more realistic and copes better with the social-media. For instance, a user might belong to: a family, one or several groups of friends, and hobbies related groups. More precisely, the overlapping community detection is either fuzzy or non-fuzzy. With the fuzzy community detection, the membership of a user in each community is weighted. For illustration purpose, consider a sport club, the involvement level (*ie.* the weight) of the oldest members in this community is higher than the one of a newcomer. On the contrary, in the non-fuzzy case, the membership is binary. Hence a user is either in or out a community. According to Xie *et al.* [161] the majority of the work proposed solution are non-fuzzy. One of the most popular method, in this line of inquiry, is the one proposed by Palla *et al.* [124].

In the *ego* based definition, one considers the communities with respect to a single user, based on a local knowledge of the user's neighborhood. This definition is proposed to study overlapping communities with a limited computational cost. This definition fits well to the social-media with partially known user-networks. This situation is usual, as discussed in Section 1.2.4. This approach is adopted, among other, by Clauset [38]. Clauset adapts the modularity measure to the local knowledge situation, and build a community "around" the node of interest, that is named the *ego*. The community is built by maintaining several sets that represent: the *ego* community; the boundary of the *ego* community; and the remaining of the other nodes that are known so far. The observed nodes are merged in the community such that the local modularity measure is maximized in a greedy way. Several updates of this approach were proposed, for instance by Bagrow *et al.* [17] or Chen *et al.* [36].

The growth of social media highlighted several problems in this research area. For instance, how to scale algorithms to gigantic user-network. Indeed, a typical social-media might have several millions nodes, and the most popular social-media count hundreds millions nodes and billions of edges. On one hand, *ego* based community detection tackles this problem by focusing on local sub-graphs. On the other hand, methods have been proposed to scale overlapping and disjoint community detection methods. Examples of the latest work in *ego* community formalization are: Ngonmang *et al.* [121], Danish *et al.* [46] or Zhou *et al.* [171]. Danish *et al.* rather than optimizing a quality measure, define a similarity measure that is based on the spread, from the *ego*, of an "opinion" that can be though of as a level of membership to the *ego* community. They study the decrease of this measure to determine the limits of the community. They furthermore propose to study communities defined with respect to multiple *ego*. Zhou and coauthors propose a generative model that has similarities with the latent dirchilet allocation [25]. This model takes into account the network and the content that each user produces, in order to produce content-based *ego* communities. The latest work in the overlapping community formalization are proposed by Yang and Leskovec [164], and Cosica *et al.* [43]. Yang and Leskovec observe that the overlap of communities are densely connected (*eg.* the more hobbies one shares with someone the more likely they are to be friends). Authors use a non-negative matrix factorization [100] that takes into account this observation. Their method is named BigCLAM. Several overlapping community detection methods are based on a non-negative matrix factorization [155, 170]. On the contrary, Coscia *et al.* use *ego* communities and label propagation [134] in a greedy optimization approach where the members of an *ego* communities "vote" for the communities attributed to the *ego*. Although using *ego* communities, this approach is based on the knowledge of the whole network. Therefore this approach complies to the overlapping community formalization. The latest work in the disjoint community formalization is proposed by Prat-Pérez *et al.* [132]. They use a quality measure based on triangles of nodes that they defined earlier [131]. They provide a large comparison to several state-of-the art methods, both in disjoint and overlapping formalizations: BigCLAM, Louvain [26], and OSLOM [99]. Authors report quality improvement with respect to the normalized mutual information [98] and more scalable results than the previous studies. Their test includes several samples from social-media such as: Youtube, Ortkut, LiveJournal, and Friendster the larger one. The evaluation is made with respect to ground-truth communities that are observed in social-media. Such ground truth communities are for instance: an *alumni* group, or a co-worker group. These groups are explicitly created and joined by

social-media users.

Another problem is to take into account how communities evolve through time. This line of inquiry is reviewed in the book chapter by Ayanaud *et al.* [16].

1.2.3 Information diffusion and social-media

The study of the spread of a user-generated-content or more generally of an information, might be done with an information diffusion model. Such models aim at define how an information circulates through the user-network, from a user to another. These models are used in various problems. For instance, the problem of the selection of the most influential users. In this problem one has to choose the user-group which will maximize the diffusion of an information. Among others, the information diffusion models can be used to infer the global outcome of a diffusion. This inference is a form a activity-prediction. This section outlines two broad lines of study of information diffusion in complex network: contagion models, and linear threshold model.

The Linear threshold models, hereafter abbreviated LT, are now presented. This class of models was devised in sociology, with the seminal work of Granovetter [71]. Before presenting the LT models, specific vocabulary is introduced. During the diffusion of an information, a user of a social-media is referred to as *active* when he-or-she participates in the diffusion (*eg.* emits a tweet about an ongoing sport event). On the contrary when this user does not participate to the diffusion, he-or-she is referred to as *inactive*. In LT models, a user becomes active when a sufficient quantity of his-or-her neighbor is already active. This mechanism is proposed to represent the "social pressure" that steers users to conform to what they perceive as the majority behavior, for instance, to relay an information. Formally, in most of the LT models, a user u has an activation threshold θ_u, and another user v influences u with strength $s_{u,v}$. The activation function of the user u is named $A(u)$ and is binary thus $A(u) \in \{0, 1\}$. The activation function is defined with respect to the set of its neighbors that is referred to as $\Gamma(u)$. The activation function is defined as follows:

$$A(u) = \left[\sum_{v \in \Gamma(u)} A(v) * s_{v,u} \right] \geq \theta_u$$

As the output of the activation function depends on the activation of the other users, the activation values might be updated until the reach of a stable situation, or stopped

before. Thus, the activation function could be defined with an additional parameter that represent the current step of the diffusion. It is noteworthy that the LT models, under specific conditions, may result in the activation of each and any user. Such a situation is named a endemic diffusion.

The contagion models constitute another class of information diffusion models, which was originally devised in the epidemiology research-area. The readers interested in a complete description of these models are referred to the book by Brauer and Castillo-Chavez [31]. We focus here one particular contagion model: the independent cascade model [68], hereafter abbreviated IC model. This model is frequently considered in the studies of the information diffusion applied to social-media. The IC model is based on a simple principle. Informally, each node might be activated by one of its neighbors. When a node u becomes active, it tries to activate each of its non-active neighbors. Consider that user u tries to activate it neighbor v, if the attempt fails the link between u and v is deemed closed for the remainder of the diffusion. That is to say, the information will not pass through this link during this diffusion episode. However others neighbor of v might try to activate v afterwards. Typically, in this problem the links are associated to weights that represent the probability for the activation attempt to succeed. Figure 1.3 illustrate this process on a toy-example situation.

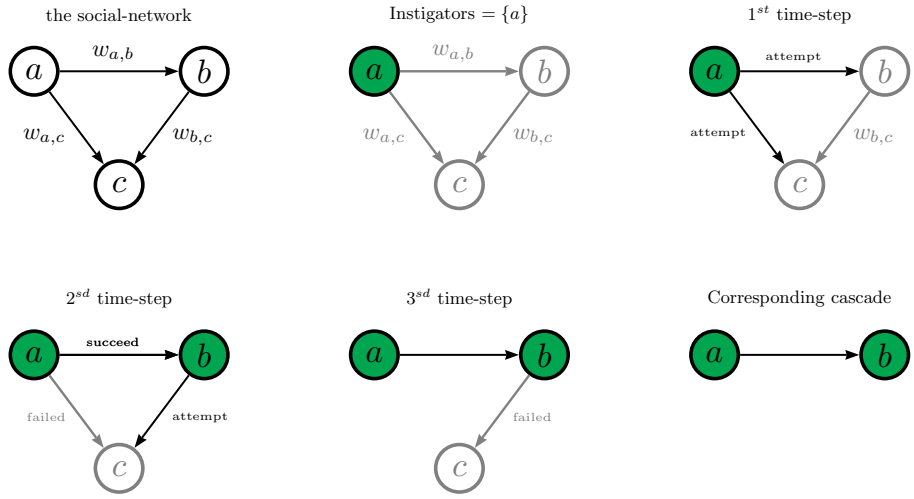

Figure 1.3: This Figure illustrates the successive steps of an information diffusion modeled as an independent cascade process over a network with three users. The edge-weight $w_{x,y}$ is the probability for user x to activate the user y when x becomes active.

Numerous updates for the IC model were proposed, some of them are presented below. Saito *et al.* [141] proposed an asynchronous version of the IC model named ASIC. The ASIC is a continuous-time model that allows a delay between the activation of a node and its attempts to activate its neighbors. In the same vein, Gomez-Rodriguez *et al.* [70] models the delay before the activation attempt with several probability distributions. Finally, a generalization for the IC models is proposed by Kempe *et al.* [88]. In this generalization, the activation function takes into account the previous activations attempts that failed. The objective is here to represent the social pressure: the more of one's neighbors tried to activate it, the more he-or-she is likely to be activate. Thus, activation function should be increasing with respect to the number of activation attempts. Moreover, it is supposed that the activation function is independent of the order of the activation attempts. Kempe *et al.* propose furthermore a generalization of the LT models, and show equivalences between these two generalizations.

The latest research effort in the prediction of the information diffusion are follows: Lagnier *et al.* [95, 96] leverage the match between the interest of the users and the content to be spread, and also take into other specific user characteristics such as their individual willingness to spread information. Another line of inquiry, proposed by Bourigault *et al.* [28] is to learn a latent space that is the best possible surrogate for a user-network which can not be entirely observed, this task is referred to as "network embedding learning".

1.2.4 Data collection from social-media

To collect social-media data is a non negligible part of the research effort in *social-media-mining*. Indeed, a large majority of these data are privately held. The most efficient way to collect data from a social-media is to use an application programming interface, which is usually referred to as an API. The API is provided by the organization that operates the social-media, thus it might be unavailable for some social-media. Each API has rules, these rules prevent malicious usages (*eg.* Spammers), enforce privacy preferences and enable companies to monetize the data. Another way to collect data from a social-media is to query directly the social-media through the Web by issuing HTTP requests. Nonetheless this approach is much less efficient than the use of the API. Moreover, most of the programming languages have dedicated libraries that overlay the API and ease their usage. Consider for instance, the python programming language [138]. There are

python-libraries for any of the best-known social-media: Twitter with `twitter`[14] library, Facebook with `facebook-sdk`[15] library, and LinkedIn with `python-linkedin`[16] library. Readers interested in further technical information are referred to the book by Rusell [140]. This book presents extensively the application programming interfaces and the libraries for Twitter, Facebook, LinkedIn and Google+. The rules enforced by the API prevent one from arbitrarily sampling the social-media. To bypass these rules might be feasible, but is not always functional. For instance, Gabielkov proposes a method to get the complete user-network from Twitter [60]. This method based on a distributed crawler, took four months to complete the data collection. In our industrial situation to rely on such a solution is not acceptable as it is not perennial.

In most of the popular social-media, the user-network cannot be fully observed. Hence, numerous studies dedicated to popular social-media are based on samples of their user-network. An important matter is then the control of the sampling quality. Few studies address this matter for popular social-media. For instance, Yong-Yeol *et al.* [13], use the fully observed user-network of Cyworld to evaluate the quality of the "snowball-sampling". This well known sampling method is dedicated to graphs. A "snowball-sampling" amounts to select a random node and retrieves its neighbors, and then their neighbors, and so on, and so forth. This breadth-first search stops when the desired sample size is reached. Yong-Yeol *et al.* compare the fully observed user-network and snowball samples of itself. They show that, in the samples, the distributions of node in-degree are also approximated by power-laws distributions. However, the coefficients of these power-laws are lower than those computed for the whole user-network. Hence, they confirm the earlier results of Lee *et al.* [101]. They furthermore give minimal sampling levels that ensure to match the full user-network characteristics. A comparable study devoted to Facebook is proposed by Gjoka *et al.* [66]. However in this study the user-network was not fully observed. Instead authors achieved a rejection sampling on user-identifiers [62]. This sample is not biased by the structure of the user-network. Gjoka *et al.* use this sample to propose several sampling methods with low bias. These sampling methods are based on adjusted random walks. To perform a rejection sampling isn't possible anymore.

Some of the essential notions and applications related to the graph theory were pre-

[14]Ressource available at `http://goo.gl/6iRa2p`
[15]Ressource available at `http://goo.gl/pzk3Yf`
[16]Ressource available at `http://goo.gl/a2MzGR`

sented. Moreover, it was shown that the user-networks are not generated by a random process. Instead, the general structure of the user-networks would generally involve: a core of "popular users" that constitute a "glue" for numerous users groups of moderate size. Moreover, the pitfalls related to the sampling and the collection of the user-networks were discussed. The next section introduces machine learning and presents how it can be used in social-media.

1.2.5 Applications of machine learning to social-media

machine learning is dedicated to automatically (*ie.* without human action) induce new knowledge from experiences. In other words, machine learning is devoted to the study, and the proposal, of computational methods that are able to improve their performance (*eg.* to decrease the forecasting error) as they are fed with new examples. The tremendous research effort in this domain is partially motivated by the "data deluge" observed for the past decade. Indeed, the data is now inexpensive to capture, store, or exchange. Thus, various stakeholders started to systematically capture data in the hope to leverage value from it. These stakeholders are for instance: companies (*eg.* transactions history, usage logs), or public authorities (*eg.* taxi GPS locations, public transport usage levels). Now, even the individuals became massive data producer. Typically, individuals produce user-generated-contents and share them through social-media. This section presents machine learning research area from a practitioner point of view. The first two parts of this section introduce the general framework for supervised and unsupervised machine learning, respectively. Finally, a supervised algorithm is presented. This presention illustrates several pitfalls encountered with social-media data. Several reference work provide a comprehensive introduction to the concepts mentioned in this section [24, 77, 117, 115]. For the remainder of this section the examples are assumed to be independently and identically distributed, as for instance the outcomes of several dice rolls. Murat *et al.* [50] present machine learning outside of this postulate.

Supervised machine learning.

The supervised machine learning has count-less applications to social-media data. Consider for instance Twitter, this social-media is used to share political information. However, in Twitter, one has not any standardized way to declare his-or-her political stance. It has been shown experimentally [126, 40] that is possible, yet not easy, to use supervised machine learning in order to infer one's political stance based on the contents

that he-or-she produces. Undoubtedly, a politician could leverage such an information to spread positive views of its political program. Vocabulary dedicated to supervised machine learning is now defined.

A textbook example of supervised machine learning problem is as follows. Knowing the weights, heights, and genders of thousands of individuals, estimate the gender of an individual that weights 85 kilos and is 1.85M tall. Figure 1.4 illustrates this problem. To answer this question, one induces a relation that relates weight and height to the gender. In other words, one is provided with sample observations of a function inputs and their subsequent outputs. One has then to propose a model that best approximates the function that was sampled. Therefore, supervised machine learning deals with structured data. Typically, the data is presented as pairs which are named instances. Each instance contains a vector of numerical or categorical values, which is referred to as feature vector (*ie.* the input of the function to be approximated). The second element of an instance is a label (*ie.* the output of the function to be approximated). The feature vectors are defined in the feature space, it is referred to as $\mathcal{X}$. The labels are elements of the label set, it is referred to as $\mathcal{Y}$. Therefore, the function to be approximated is of form $f : \mathcal{X} \mapsto \mathcal{Y}$.

When labels are drawn from a finite set of values, the associated problem is referred to as a *classification problem*. More precisely, when there is exactly two labels, the problem is referred to as a *binary classification problem*. In a binary classification problem the function to be approximated is a decision function. The gender estimation problem, as presented earlier, is a binary classification problem. On the contrary, if there are more than two classes (*eg.* male, female, and undisclosed), then the problem is referred to as a *multi-class classification problem*. Figures 1.4 and 1.5 present both of these problems, in a feature space that has two dimensions (*eg.* the weights, and the heights).

When the labels are continuous, the problem is referred to as a *regression problem*. An example of a regression problem is to predict the blood pressure, in Millimeter of mercury, of an individual while knowing the individual's height and weight. This problem is illustrated in Figure 1.6.

Supervised machine learning, may it be as a classification or a regression problem, can be considered in several application context. Such a context is referred to as a task, the best-known tasks are as follows.

- In batch learning, one receives n instances $\{(\mathbf{x}_1, \mathbf{y}_1), \ldots, (\mathbf{x}_n, \mathbf{y}_n)\}$. This set of instances is usually referred to as the *training-set*. These instances are used to define (*ie.* fit the parameters) the function $\hat{f} : \mathcal{X} \mapsto \mathcal{Y}$ which approximate f. This step is usually referred to as the *training*. Recall that f is a function that relates $\mathbf{x}_i$ to $\mathbf{y}_i$. Thereafter, the function $\hat{f}$ can be applied on a, yet unseen, set of feature vectors $\{\mathbf{x'}_1, \ldots, \mathbf{x'}_m\}$ in order to approximate the corresponding unknown labels $\{\mathbf{y'}_1, \ldots, \mathbf{y'}_m\}$ with $\{\hat{f}(\mathbf{x'}_1), \ldots, \hat{f}(\mathbf{x'}_m)\}$. The set of feature vector whose label is unknown is usually referred to the test-set. To apply the function $\hat{f}$ is usually referred to as the *labeling*. It exists numerous measures that estimate the quality of the function $\hat{f}$. These measures are based on a comparison of $\hat{f}(\mathbf{x'}_i)$ and $\mathbf{y'}_i$. These measures depends on the considered problem. For instance, in binary classification problem one might use the area under the receiver operating characteristic curve. In a regression problem, a standard measure is the root mean square error [24, 18].

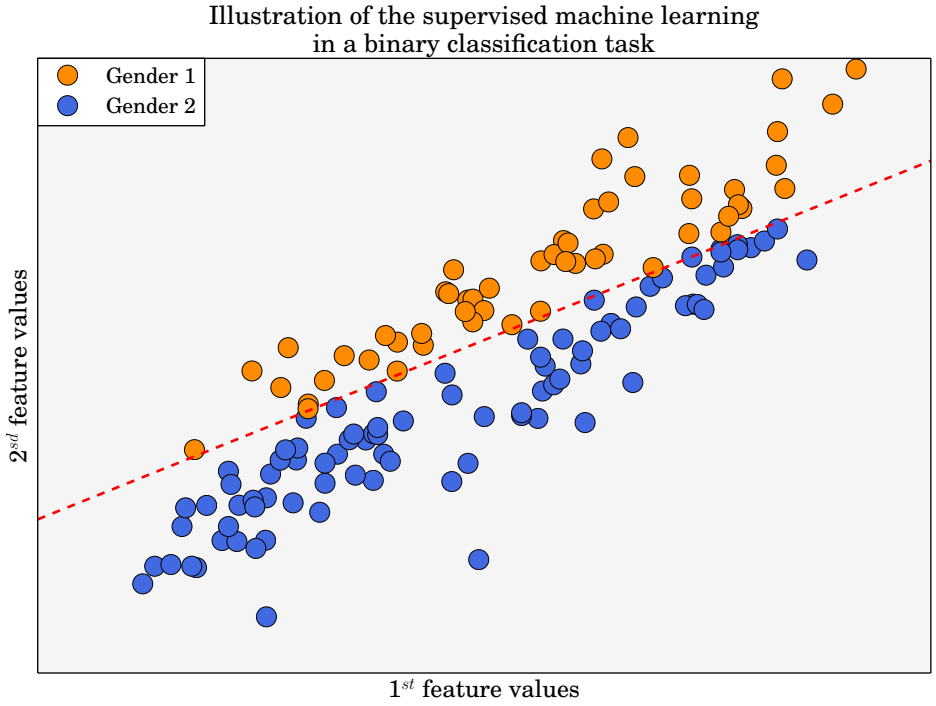

Figure 1.4: This illustration present a binary classification that is performed on two features. The features are here represented as the two dimensions of the plan. These features could be the weight and the height of individuals, for instance. Therefore each point in the plan represents the weight and the height of an individual. In addition, the color of each point represents the individual's gender (*ie.* the label).

- In online learning, one first receives a feature vector: $\mathbf{x}_i \in \mathcal{X}$ and then proposes a corresponding label, referred to as $\hat{f}(\mathbf{x}_i) = \hat{\mathbf{y}}_i$. Later on, the true label, which is referred to as $\mathbf{y}_i$, is received. The true label is used to update the function $\hat{f}$ in order to reduce the difference between the approximated label: $\hat{\mathbf{y}}_i$, and the true label: $\mathbf{y}_i$. This pattern is repeated for each new feature vector that is presented. Typically, at the time of the first prediction, the function $\hat{f}$ is either random or reflects a prior knowledge about $\mathcal{Y}$. The online learning is also known as incremental learning.

- In semi-supervised learning, one receives n feature vectors $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$, and the subsequent labels $\{\mathbf{y}_1, \ldots, \mathbf{y}_n\}$. In addition, m others feature vectors are available $\mathbf{X}' = \{\mathbf{x}_{n+1}, \ldots, \mathbf{x}_m\}$ but the subsequent labels are unknown in this case. Supervised machine learning methods are applied to features vectors in $\mathbf{X}$, and unsupervised machine learning methods, as for instance clustering, are applied to features vectors in $\mathbf{X}'$. More precisely, when one aims at predict only the labels associated to $\mathbf{X}'$ the task amounts to transduction [61].

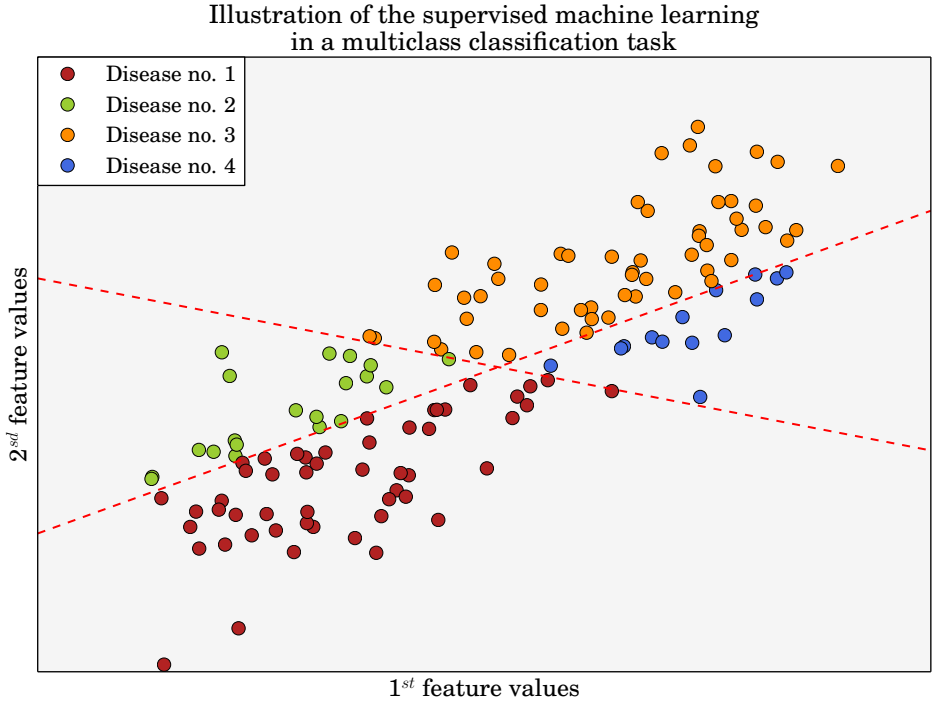

Figure 1.5: The multi-class classification, a supervised machine learning task, is here presented on synthetic data. This multi-class classification is performed with respect to two features that are represented as the two dimensions of the plan. The color of each point represents the label value, here a disease to be diagnosed.

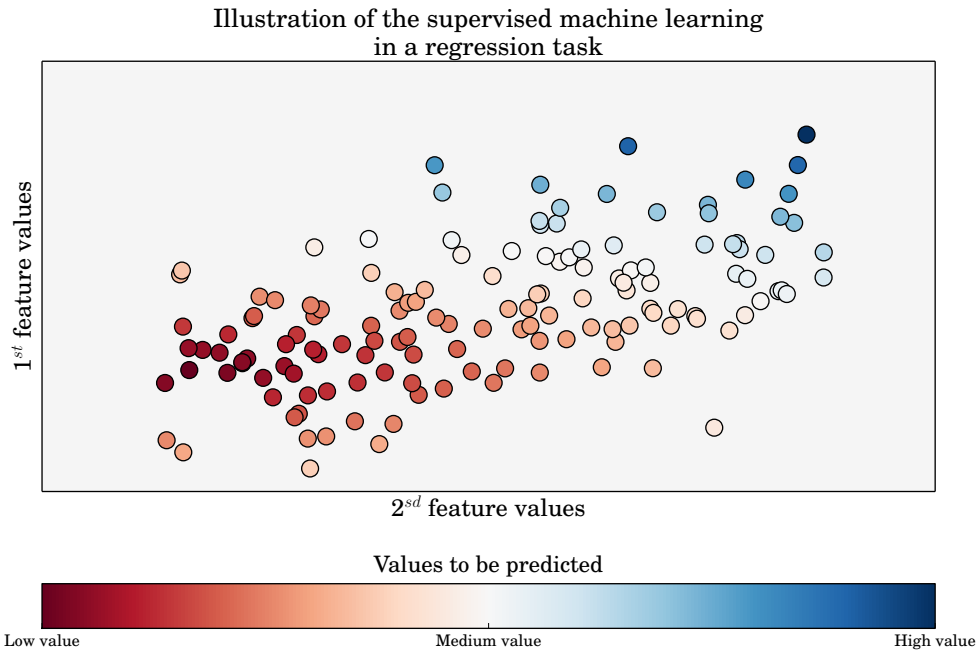

Figure 1.6: The regression, a supervised machine learning task, is here presented on synthetic data. This regression is performed with respect to two features that are represented here as the two dimensions of the plan. The color of each point represents the label value in $\mathbb{R}$.

It is noteworthy that the supervised machine learning is here presented in its discriminative approach. Indeed, this approach designates all the methods which directly model the posterior probability of the label, knowing the feature vector. The discriminative approach is opposed to the generative one. In the generative approach, the distribution of the feature space is first modeled, then posterior probabilities of each label, knowing a feature vector, are used to choose a label.

Unsupervised machine learning.

In a unsupervised machine learning task, one receives the features vectors $\{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$, but contrary to the supervised machine learning, the labels are never observed. Therefore, the goal is here to represent the observations in a way that ease the reasoning and the formulation of new knowledge.

The best known task in unsupervised learning is the clustering. The clustering methods are extensively described in several works [81, 23]. In this task one has to propose a

partition of the feature space. In this partition, the elements that are the most similar to each others, should belong to the same part. Each part is named a cluster. As an illustration consider a feature space that describe mammals. An acceptable clustering is as follows: marine mammals, land-based mammals, and finally flying mammals. However, the key difficulty in clustering is that it may exists multiple, equally interesting, partitions of a single feature space. In the same example, an equally interesting partition is as follows: carnivores, omnivorous and herbivores. This situation emphasize the importance of the distance function used to compare the objects. The definition of a distance is as follows.

Definition 1 *Distance function.* $\forall\ x_i, x_j, x_k$*, the function* d *is a distance function if it verifies four properties:*

1. *Symmetry:* $d(x_i, x_j) = d(x_j, x_i)$
2. *Non negativity:* $0 \leq d(x_i, x_j)$
3. *Coincidence axiom:* $x_i = x_j \Leftrightarrow d(x_j, x_i) = 0$
4. *Triangle inequality:* $d(x_i, x_k) \leq d(x_i, x_j) + d(x_j, x_k)$

For instance, the Euclidean distance is the most intuitive distance in our everyday life experience. This distance belong to the family of the Minkowski distances and has parameter $p = 2$. In a space of dimensionality n, the Euclidean distance is defined as:

$$d(\mathbf{x}, \mathbf{x}') = \left(\sum_{i=1}^{n} |\mathbf{x}_i - \mathbf{x'}_i|^2 \right)^{1/2}$$

The important research effort devoted to clustering testify of the importance and the complexity of this question. When one processes social-media data, the clustering is generally used to group user-generated-contents into several topics (*eg.* sports, science, politics, and so on). The clustering might also be applied directly to users, but it in most of the case a user is described, at least partially, by his-or-her contents.

Usage of the supervised machine learning.

In order to apply machine learning to a practical situation, one has to avoid several pitfalls which are presented below. For illustration purpose, consider a binary classification problem in a batch learning task. The problem is as follows: in Twitter, label

each tweet with the gender of his-or-her author. Assume that this problem is tackled with the k-nearest neighbors algorithm, from now on abbreviated k-NN [56]. k-NN is a supervised learning algorithm suitable for classification or regression problems, it is now presented. k-NN takes as input n instances $\{(\mathbf{x}_1, \mathbf{y}_1), \ldots, (\mathbf{x}_n, \mathbf{y}_n)\}$. The objective is then to approximate the label $\mathbf{y}' \in \{-1, +1\}$ associated to $\mathbf{x}'$ a yet unseen feature vector. The k-NN classifier consists of two steps:

1. To compute the distance (*eg.* the Euclidean distance), between $\mathbf{x}'$ and every feature vector in the training-set, that is to say $\{d(\mathbf{x}', \mathbf{x_1}), \ldots, d(\mathbf{x}', \mathbf{x_n})\}$.

2. To output $\mathbf{y}'$ so that it is equal to the majority label among the k closest neighbors.

Figure 1.7 illustrates a k-NN output. This algorithm can be updated to weight the contribution of each neighbor. Typically, the weighting function is based on the distance to the neighbor and may reflect a specific knowledge about the application domain. If the labels are continuous (*ie.* a regression problem), the `majority` function is replaced by a weighted average. It is noteworthy that k-NN has no parameter to tune, therefore it has not any training step, and all the computational cost lies in the labeling. On the contrary, in most of others algorithms the computational cost lies in the training step. In order to propose a solution for the considered problem, a general pattern is as follows:

1. One defines precisely the properties of the domain of interest. In this situation, identify which information can be leveraged from the tweet. For instance, the content of the tweet, the user name, or the user picture.

2. One defines the features with respect to the information that were identified previously. The features represent all the actionable knowledge for machine learning algorithm. Therefore, they have to be defined cautiously. The features can be used to encode a prior knowledge about the domain of application. In this situation, a prior knowledge is that some first names are typically masculine, or feminine, or both (*eg.* Taylor, Jordan). Hence, one may use the data from the national statistics office and compute the distribution of gender per first name. With this information, one may define a new feature. This new feature might replace, or enrich, the "raw" user first name.

3. One builds the training-set, that is to say collect the feature vectors and the subsequent labels. This step might be time consuming, and has to be done meticulously. If the labels are not available one has to produce them, for instance with human

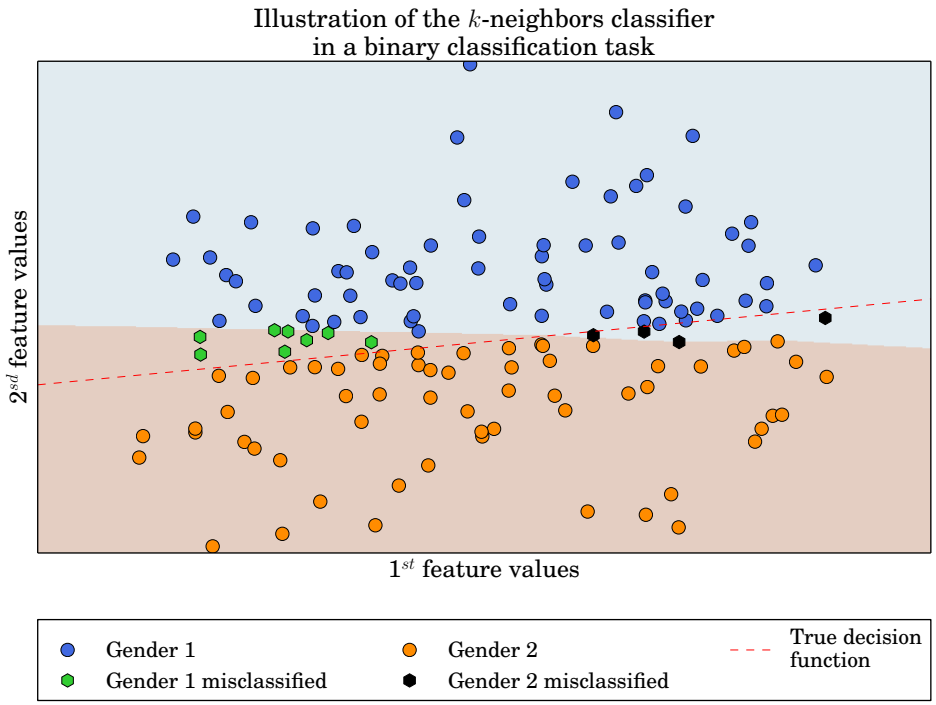

Figure 1.7: In this illustration a circle represents a correctly classified point, and an hexagon represents a misclassified point. The red dotted line represent the decision function that is unknown and has to be approximated. Moreover, the background color represents the approximated decision function. This instances of the train-set are not displayed in this Figure.

annotators. This substantial burden might be distributed through a crowdsourcing marketplace, as for instance amazon mechanical turk[17] [125].

4. One studies the collected data, and the domain of interest, in order to choose a proper machine learning algorithm. In this situation, one expect to have large training and testing sets, with respectively n and m elements, for instance. Therefore k-NN, whose complexity is in $\mathcal{O}(nm)$ for the labeling step, would not be the better choice. Note that it exists several speed-up methods for k-NN, as for instance the ones based on a ball-tree partition of the feature space [105]. In such a situation one could prefer an algorithm whose computational cost lies in the training rather than in the labeling, as for instance one based on ensemble methods [172]. In addition, one has to consider the types (*ie.* continuous or discrete) and characteristic of each feature (*ie.* variance, median, and so on). Indeed some algorithm need for

[17]Ressource available at `http://www.mturk.com`

the features to share the same scale. For instance, the k-neighbors algorithm, when used with unscaled features and the Euclidean distance, would mostly take into account the feature with the largest variance.

5. One selects the model that has the better performances. That is to say, to tune the meta-parameters in order to obtain the best approximation of the labels. This task is usually referred to as *model selection*. In this situation, k-NN has three meta-parameters: the number of neighbors to be considered k, the distance function, and the weighting function. Generally, a part of the training-set-labels are deemed unobserved, and the corresponding feature vectors are used as test-set. This test-set is used to evaluate several combination of the meta-parameters. The selection of the meta-parameters has to be done with respect to the properties of the considered algorithm. For instance with k-NN, the selection of the meta-parameter k implies different bias-variance trade-off. Indeed, to consider few neighbors may lead to an high variance (*ie.* unstable approximation). On the contrary, to consider a large number of neighbors may lead to a biased model (*ie.* with poor results on unseen data) [63].

This iterative process might be repeated several times for a given problem. Indeed, one acquires new knowledge during the process. These new knowledge might be used to improve the performances of the proposed solution.

The general framework for machine learning was presented along with a detailed application example. This example illustrated several pitfalls encountered in the use of supervised machine learning methods with social-media data. The previous sub-sections described social-media and key questions related to activity predicition in social-media. The next section structures the considerable research effort devoted to the prediction of the activity in social-media. To this end, key concepts are presented. Afterwards, selected studies are discussed.

1.3 Activity-prediction and related questions

The broad usage of social-media produced an unprecedented amount of data. These data might be leveraged to tackle numerous problems, as for instance: to predict the life expectancy of an online community [84]; to predict the purchases behaviors [169]; to predict companies merges and acquisitions [160]; to predict venues of interest [19];

to predict box-office revenue [14]; or to predict influenza like illness rates [45]. Another problem, that can be tackled within any social-media, is to predict the activity of a topic or thematic. We refer to this problem as the activity-prediction problem. To propose a solution to this problem has numerous industrial usages. For instance, the company which funded this thesis, plans to predict the activity for each product of its product database (*eg.* iphone, ipad, google glass), in order to drive the content production of the editors, and to prepare deals with online retailers. The activity-prediction remains an open question, as can be seen from the recent publications conference devoted to the data-mining and the web, as for instance: WWW, ICWSM, WSDM, KDD. The presentation of this still-growing research area, is structured as follows. Firstly the concepts that are the most frequently used are presented. Secondly several taxonomies for the spikes of collective attention are reviewed. Afterwards the two lines of inquiry that are the most germane to our work are presented.

1.3.1 Notions related to activity prediction

Several concepts are frequently used in the publications related to the activity-prediction problem in social media. These concepts are summarized and formally defined in this section. This summarization structures the work effort dedicated to activity-prediction. The recurrent concepts are related to: the user-generated-contents, the topics, and the activity measures. According to the presentation of a social-media in Section 1.1, one who observe a social-media collects user-generated-contents through time. The set of collected user-generated-contents is named $\mathbf{C} = \{\gamma_1, \ldots, \gamma_k\}$. It is common to use a discrete time representation. When one uses a discrete time representation, the observations are grouped in several time-periods, that are defined as follows.

Definition 2 ***Time-period.*** *When the time is discretized, one considers that the observations are distributed in a sequences of contiguous time-periods named* $\mathbf{T} = \{t_1, \ldots, t_{max}\}$. *A time-period* t_i *begins at the time* b_i *and ends at the time* e_i. *Therefore, every user-generated-content that is produced after* b_i *(inclusive), and before* e_i *(exclusive), belongs to* t_i. *The time-periods are contiguous, thus* $b_i = e_{i-1}$ *and* $e_i = b_{i+1}$. *For the sake of concision the subscript might be dropped, then a time-period is simply named* t.

The topic is a key concept related to activity-prediction in social-media. The definition of a topic varies from a study to another. There are three definitions for the topic concept. These three definition are considered interchangeable, as suggested by Guille *et al.* [75]. These three definitions are presented below.

Definition 3 ***Topic.*** *Consider a user-generated-content that has a textual content. That content is divided in a sequence of tokens. In this situation, a topic is either a: (a) token; (b) sequence of tokens; (c) probability distribution over the tokens of a known vocabulary. A user-generated-content refers to a topic z, if it contains the topical token(s) of z, or if its tokens are distributed according to the topic z. The set of topics is named* $\mathbf{Z}$*, it contains all the topics* $\{z_1, \ldots, z_m\}$*. For the sake of concision the subscript might be dropped, then a topic is simply named* z*.*

A token that defines a topic is associated to a single thematic. For instance the token "iPhone" is associated to the "apple smartphone" thematic. On the contrary tokens such as: "a","of" or "then" aren't associated to any specific thematic. These irrelevant tokens are usually ignored. New tokens are frequently created, indeed user-generated-contents contains misspelled words as well as new spellings. Moreover, in some popular social-media such as: Twitter and Instagram, a widespread convention is to prefix with # the tokens that carry a specific thematic. For instance, during the "*Occupy Wall Street*" protest, the token #OWS, was used to refer to the protest. These tokens are named "hashtags". An hashtag might be created from a concatenation of tokens. For instance, #MusicMonday is used to publicize a song on Monday. The user-generated-contents can be grouped with respect to the topics in order to define the Content-activity.

Definition 4 ***Content-activity*** *The set of user-generated-contents that are observed in the social-media is divided in subsets with respect to the topics. Each subset describes one single topic and contains every user-generated-content that refers to this topic. Consider a topic named z, the set of user-generated-contents that refers to z is defined as $\mathcal{C}_z$. When the time is discretized in time-periods, the set of user-generated-contents which refer to a topic named z that are observed during the time-period t is defined as $\mathcal{C}_z(t)$. The amount of user-generated-content is defined with $|\cdot|$ the set-cardinal operator. The amount of user-generated-content is $|\mathcal{C}_z(t)|$. This amount is named the content-activity of topic z for the time period t. When the time is not discretized, the content-activity of topic z is $|\mathcal{C}_z|$.*

When the content-activity of a topic increases significantly during several consecutive time-periods and then falls back to usual values, the topic is said to "burst". An informal definition of a activity-burst is as follows.

Definition 5 ***activity-burst.*** *Consider the time-series that describe the content-activity of the topic z, that is defined as follows: $X_{z,i} = |\mathcal{C}_z(i)|$. Consider in addition, two*

thresholds: σ and θ. In such a situation, a topic z is said to have a activity-burst from time-period t to time period t', when $\tau \notin [t, t'] \Rightarrow X_{z,\tau} < \sigma$ and $\tau \in [t, t'] \Rightarrow X_{z,\tau} > \sigma * \theta$.

A topic that bursts and which content-activity is among the top-n in the whole social-media, is said to be the object of a spike of collective attention. A spike of collective attention is also referred to as a "trending topic" in Twitter, or more generally to a viral diffusion.

In the remaining of this section two lines of inquiry for the activity-prediction are presented. Firstly the prediction of the activity of one single **user-generated-content**. As an illustration consider Twitter where a user-generated-content is a Tweet. In such a case, to predict the activity of one single user-generated-content amount to predict of the number of re-tweets, or replies, that one Tweet will get. On the contrary, the second line of inquiry, is devoted to predict the **topic-activity**. The activity of a topic named z is based on the set of user-generated-contents $\mathcal{C}_z$ at the time of interest. When the time is discretized in time-periods, the activity measure is generally based on $\mathcal{C}_z(t)$ with $t \in \mathcal{T}$ the time-period for which the prediction is done. In order to illustrate the difference between these two inquiry lines, consider the situation that follows: a breaking news topic named $z \in \mathcal{Z}$ spreads across a social-media. An influential user emits one user-generated-content, that refer to z. This user-generated-content gets relayed by m other users. In the meantime, n other users produce spontaneously (*ie.* without relaying) one user-generated-content each. These n user-generated-contents refer also to z the breaking news topic. The prediction of **user-generated-content** activity amounts to approximate a function f such that $f(c) \approx m$. On the contrary the **topic-activity** prediction amounts to approximate a function $f'(z) \approx 1 + n + m$. For the sake of clarity the temporal questions are not mentioned in the above example. However the functions f and f' are most of the time considered to be time-dependent. These two lines of inquiry are are complementary as illustrated by Yang *et al.* [165]. Authors pointed out the differences between the prediction of a spontaneous user-generated-content and the prediction of sharing user-generated-content that already exist. In addition, empirical studies [94, 165] observed that an important share of the user-generated-content are spontaneous. To describe accurately the social-media the spontaneous creation of user-generated-content has to be taken into account.

Despite these two lines of inquiry, the study of activity in social-media is often summarized to the study of the spikes of collective attention, although by definition theses

spikes affect few topics. Moreover, the spikes of collective attention do not represent of the whole activity in the social-media. The study of the spikes of collective attention received a great deal of attention. For instance, several empirical studies [94, 163, 102] reveal that spikes of collective attention have specific characteristics. These studies are now reviewed.

1.3.2 Taxonomy of the spikes of collective attention

The study of "spikes of collective attention" received a great deal of efforts. Indeed to leverage these phenomenons has numerous applications including viral marketing. Twitter provides a list of "*trending-topic*" which are the current spikes of collective attention. Although often studied in Twitter, these spikes occurs in any social-media. For instance, Leskovec *et al.* [103] studied attention spikes and their respective effects on broadcast media such as CNN and weblogs. As we shall illustrate, several studies devoted to social-media, and especially Twitter, conclude that the spikes of collective attention can be described with a limited number life-cycle-patterns. Therefore, a taxonomy of these spikes of attention can be devised.

In their work Kwak *et al.* [94] study 4226 unique *trending-topics* (hereafter abbreviated TTP), and use time-periods that cover one day. They note that 20% of the users participated in at least one TTP during their observation. Authors compared the TTP to the trends reported in Google and to the headlines of CNN. It appears that the trends are more versatile in Google. In addition the TTP that match a CNN headline were, more than half time, reported first in CNN. The news that first appear in Twitter are generally related to live broadcast event (*eg.* sports), these results are confirmed by Petrovic *et al.* [128]. Another interesting point to consider, is the way that users participate in a TTP. More precisely, which are the proportions of: re-tweet; reply; mention (*ie.* a mention is a tweet directed to a user); and simple tweet (*ie.* none of the aforementioned cases), observed during a TTP. Authors report that most a third of the user-generated-contents during a TTP are re-tweets [18]. The majority of the messages exchanged during a TTP are spontaneously generated tweets. The life cycle of a TTP is also studied. A TTP is considered inactive when, during one whole time-period, not any tweet refers to the TTP. The vast majority (73%) of TTP disapear after their first period of inactivity, 15% after the second period of inactivity, and 5% after the third period of activity. Finally, a

[18] excepted, according to the authors, a probable bug with 80% of re-tweet observed

categorization of the TTP is proposed, this categorization is based on a model proposed by Crane and Sornette [44]. This categorization takes into account two factors: the type of event that underlies the TTP, and the "*ability of individuals to influence others to action*" which is discretized in two classes: critical (high), and subcritical (low). The type of event that underlies the TTP is either endogenous or exogenous. An endogenous event comes from the social-media, as for instance the hashtag #MusicMonday, which is used to publicize a song on Monday. An exogenous event comes from outside of the social-media, as for instance an earthquake or the release of a new iphone. The distribution among these four categories of TTP is reproduced in Table 1.2. Authors have manually

	Subcritical	Critical
exogenous	31.5% (1,905)	54.3% (3,290)
endogenous	6.9% (419)	7.3% (444)

Table 1.2: This table presents the headcount for each of the four categories of *trending-topics* reported by Kwak *et al.* [94].

inspected the topics, and outline that the exogenous critical topics are mostly breaking news. On the contrary, endogenous critical topics are "*of more lasting nature*". Such topics are associated to recurrent information to share, for instance a brand name. Table 1.2 shows that in twitter the spikes of collective attention are mostly related to breaking news.

Naaman *et al.* [118] devised a more detailed taxonomy. To this end, they observed the spikes of attention localized in New-York city. Regarding, the exogenous TTP, Naaman *et al.* distinguish those event which are: the "*Broadcast-media events*" in which the local and the global broadcast are distinguished; the "*Global news events*" in which breaking news event are distinguished from non-breaking news event. A breaking news event in an unplanned event such as an earthquake. On the contrary, a non-breaking news corresponds to planned events, such as a movie release; the "*National recurrent events*", such as Halloween or Christmas; and finally the "*Local participatory and physical events*" which are divided in the same way than the global news event. Regarding the endogenous TTP, Naaman *et al.* distinguish: Internet memes such as #MusicMonday; Popular users that are re-tweeted in mass by their followers; and Fan community.

In their work Lehmann *et al.* [102] confirm the observation of Kwak *et al.*, and propose

a more general taxonomy than the one devised by Namaan *et al.*. This study is based on a five months observation of Twitter. Within this observation period authors counted that 402 topics were subject to a spike of collective attention. In their study the time is discretized per day. For a topic named z, the spike of attention is divided in three parts: (a) the two weeks that predate the day of maximum attention; (b) the day of maximum attention defined as $d = \text{argmax}_t(|\mathcal{C}_z(t)|)$; and (c) the two weeks that postdate the day of maximum attention. A generic taxonomy is obtained by normalizing the content-activity of each part by the sum of the content-activity observed. Authors used the Expectation Maximization [117] to learn an optimal Gaussian mixture model [117] and fixed the number of component of the mixture model using the Bayesian Information Criterion. They obtain four clusters. The first contains the spikes of attention that are mostly active before the peak, which corresponds the anticipation of a planned event, as for instance Christmas. The second cluster contains the spikes of attention with a symmetric activity before and after the peak. Authors associate these peaks with "*endogenous propagation over the social network*", as for instance the release of a movie that sparks reactions even after its release. The third cluster contains the spikes of attention with most of the attention after the peak. Authors associate these peaks with unexpected events such as the "*breaking news*" described by Namaan *et al.*. The fourth cluster contains the spikes of attention with most of the attention concentrated on the day of the peak itself, these ephemerals, as per Kwak *et al.*, corresponds the exogenous subcritical topics.

Other studies are devoted to this phenomenons. For instance: Yang and Leskovec [163] propose to cluster the pattern of exogenous spikes of attention, with a time resolution of one hour; Romero [136] studied, among other, the importance of topical category (*eg.* political, sports) for the adoption a popular hashtag; Sitaram *et al.* [15] conclude that the trending topics are mostly supported by re-tweets, contrary to the observation of Kwak *et al.*. This difference might related to the differences in their sampling methods. Globally, The results presented above, confirm the existence of few patterns of life-cycle for a spike of collective attention in Twitter. In addition, these patterns depend on the event-type to which they are related. These results are promising in term of the predictability for certain types of spike. Mainly, the spikes related to a scheduled event, may it be: recurrent such a Christmas, or punctual such as the release of a new iphone. However, it is noteworthy that most of the topics does not become the object of a spike of collective attention, as mentioned by Kong *et al.* [91]. For instance, some topics alternate growths and decays as the public interest varies without ever being salient. In such a case, the

patterns of life-cycle, observed for the spikes of collective attention might not generalize well.

1.3.3 Predicting topic activity

The studies related to the topic-activity problem are now presented, to this end, the definitions presented in Section 1.3.1 are used. In these studies, the activity of a topic is defined in two different ways: (a) as a quantity of user-generated-content, or (b) as a quantity of users. The quantity of user-generated-content for a topic named z during a time-period t is the cardinal of the set of user-generated-content that were created during t and that refer to z, as per Definition 4 this set is $\mathcal{C}_z(t)$. The quantity of users that produced these user-generated-content is the number of distinct users that produced $\mathcal{C}_z(t)$. This quantity of users is referred to as $\mathcal{U}_z(t)$ for the time-period t and $\mathcal{U}_z$ when the time is not discretized into time-periods. Both of these measures are defined for one topic and a time-period. The informal definition of the topic-activity prediction problem is as follows.

Definition 6 ***Topic-activity prediction.*** *The topic-activity might be predicted for: (a) the next time-period(s); (b) until the topic ceases to be active, that is to say when the topic stops to be referred by any new user-generated-content, or (c) until the end of the observation. The prediction of the activity for a topic named z is based on the past observations of user-generated-contents that refers to z as well as other informations.*

In this research area, the state-of-art work are based on four types of information: the user-network topology; the content of the topic; the characteristics of the time-series that describe the temporal evolution of the topic; and the characteristics of the users that discuss the topic. A frequent approach is to use these information as features of a supervised machine learning problem, as presented in Section 1.2.5. Another approach is to define an information diffusion model as presented in Section 1.2.1, and to use these information to devise the probability of activation for each node. A third approach, is to model directly a quantity that is used to predict the activity, as for instance the influence of a node on the rate of diffusion [162].

It is important to stress-out that the topic-activity is not equivalent to the user-generated-content-popularity. Indeed, as shown earlier, the topic-activity prediction takes into account any user-generated-contents that refers to the topic of interest. On the

contrary, in the user-generated-content-popularity, one tries to predict how many user-generated-content will be observed in reaction to one single user-generated-content. In this prediction problem, the spontaneous content (*ie.* without any reference to an existing user-generated-content) are ignored. We now review the latest and most notable work devoted to the topic-activity prediction. In this review, the work are grouped with respect to the prediction setting that is tackled. There are several prediction settings. Consider a topic z which activity has to be predicted. This prediction might be based on: (a) the latest observed user-generated-contents about z; (b) the few first observed user-generated-contents about z. The studies that fit in these two settings are now presented. Afterwards a third setting, which is specific to the prediction of spikes of collective attention will be presented.

Topic-activity based on the latest topic-activity.

When the prediction is based on latest observed topic-activity, one considers the last few observations before the present time and tries to predict the upcoming activity. The studies that base the activity-prediction on the latest stages of the topic-activity are reviewed below.

In their work Zhang *et al.* [168] study the topic-activity prediction in a machine learning framework. Authors consider the two activity measures presented above: the number of user-generated-contents, and number of users that produce them. In this work devoted to Twitter, a topic is considered to be represented by an hashtag and a time-period covers one day. More precisely, this study is based on the 366 most popular hashtags related to the "Arab spring" [106], each of these hashtags was used at least 5000 times. This study bears similarities with the work by Romero *et al.* and Yang *et al.* which are presented below. However, in this study authors compare the importance of three types of information: the user-network topology; the content of the topic; and the characteristics of the users that spread the topic. More precisely, the user-network is observed with respect to one hashtag of interest, hereafter named z. This user-network represents the result of the diffusion of z. Indeed, the considered user-network contains only the adopters of z (*ie.* users that produced a user-generated-content which refers to z) and their followers which are the "border" of the topic-spread. The user-network, observed for the hashtag z, is divided in order to distinguish the spontaneous adopters from the adopters that re-tweeted a message of the earlier adopters of z. Regarding the

features based on the user-network topology, authors use the density and the reciprocity of the subgraphs of the user-network (as presented in Section 1.2.1). Another feature is defined as the share of adopters which belong to the "border". Several features based on the content of the topic are used. For instance, the portion of tweets that refer to z and contain a URL. This feature was firstly proposed by Bongwon *et al.* [145]. Another content based feature is the proportion of the spontaneous adopters out of the whole adopters. Finally, the features based on the adopters-characteristics, rely on measures defined for a single user that are averaged over the whole set of adopters. These features aim to take into account the will of the users to spread the hashtags, their general level of activity, and so on. The importance of each feature is determined by the random forest algorithm as proposed by Breiman [32]. This supervised machine learning algorithm, which belongs to the ensemble methods, uses a set of decision trees. Therefore the feature importance, which is computed in each tree, is averaged over the whole set of decision trees. The importance of a feature, in a specific tree, corresponds to the improvement in the split-criterion observed for this feature. Readers interested in more details about the random forest are referred to the book by Hastie *et al.* [77]. The random forest is trained with the five latest time-periods, and the forecast horizon is equal to one time-period. That is to say, one observes five days to predict the activity for the next day. In this situation, authors compare the importance of each set of feature, and compare them with a prediction based solely on the past activity observations. The conclusions are as follows. The prediction results are improved by the proposed features. The most important features are the ones that describe the adopters-characteristics. This observation holds for the two activity measures described earlier: (a) the number of adopters; and (b) number of user-generated-contents. Nonetheless the combination of the three feature sets achieve better results than any single feature set. Finally authors compare several prediction models, and note a substantial improvement with respect to two baselines. More precisely, these baselines are: (a) to predict the mean of the previous activity values, and (b) to predict the activity for the latest time-period of observation. The best results are obtained by a feed-forward neural network, the random forest obtain comparable, yet less stable, results. The prediction error are reported with the mean-squared-error measure. It is noteworthy that the prediction are done on the base-10 logarithm of the activity measures. The use of base-10 logarithm is a way to emphasize the prediction error made on the weakly popular topics, and de-emphasize the prediction error made on the most popular topics. Base-10 logarithm is also used by Tsur and Rappoport [151].

Prediction based on early stages of the topic-activity.

When one predicts the topic-activity with respect to theirs early-stage activity, it implies to discard the topics which are already active at the beginning of the observation. For instance as of 2014 the topic "Iphone" is already active as users refer to it frequently. Therefore, the activity of the topic "iphone" is not studied in this experimental setting. Topic that are usable in this prediction setting can be: (a) topics which has a cyclic activity, and (b) newly defined topic. For instance, the topic "Christmas" is likely to be active for few months in a year, and start over each year after a period of inactivity. On the contrary a topic that refers to a particular event, for instance the "Occupy Wall Street" protests begins to be used as the protests starts, and is less likely to have such a cyclic behavior. The studies that do activity-prediction based on the early stages of the topic-activity are reviewed below.

The work of Romero *et al.* [137] aims at study the interplay between to user-network and the adoption of a topic. This work is evaluated on Twitter, therefore, as in many other studies, a topic is here considered to be represented by an hashtag. The predictions are done with respect to the end of the observations. Romero *et al.* propose to predict: the link formation based on the topics that users spread, and the number of users that discuss an hashtag, hereafter named adopters. We review here the second task, where the activity to predict is the cardinal of the set of users that produced user-generated-contents which refer to topic z, namely $\mathcal{C}_z$. Authors define two user-networks: (a) The mention graph, based on the "mentions" mechanism, that is proper to Twitter. In Twitter a user named u can direct a message to user named v by adding "@u" to the message. In this graph, an edge is created from user u to user v, when the user u has "mentioned" the user v at least k times. (b) The full graph, based on the "follow" relationship. This relation is asymmetric, thus this graph is directed. It is noteworthy, that the mention-graph definition is frequently used in other work related to the activity prediction. Indeed, it is inexpensive to build, and might carry a different meaning than the network based on the follow relationship. However, several thresholds for the value k have to be tested. In this study, an hashtag that does not reach 1000 adopters is discarded. The activity-prediction is based on the user-networks that are made from the 1000 first adopters. Here, the activity-predivalction is considered as a binary-classification task. An hashtag whose adopters-quantity double is labeled as positive, otherwise it is labeled as negative. Authors tested the prediction with respect to three quantity of initial adopters: 1000,

2000, and 4000 adopters, and tested each definition of the user-network. Authors compared their approach to majority-vote base line, which label positive every hashtag if the majority of the hashtag observed in the training-set are positive. The baseline has an accuracy of 0.53. A logistic regression based on the user-networks characteristics achieve an accuracy around 0.67 when one considers the follower network. The accuracy varies with respect to the initial quantity of adopters. Authors note that none feature performs better than the combination of all the features. Finally authors conclude that the activity of the hashtag is highest when the density of the graph of initial adopters, is either very low or very high.

Another study of the topic-activity prediction is the one of Weng *et al.* [157]. In this study devoted to Twitter, an hashtag is considered as a topic. The topic-activity of a topic named z is measured as: the number of users $|\mathcal{U}_z|$, and the number of tweet they produced: $|\mathcal{C}_z|$ before the end of the observation. In this study the observation last for two months, and a topic that had more than 20 tweets during the month before the beginning of the observation is discarded. Authors tackle the activity-prediction with a *multi-class classification problem* as presented in Section 1.2.5. The features are computed with respect to the set of the n-first tweets, with n that varies in $\{25, 50, 100\}$. This novel approach allow to compute the features with respect to an unbounded time period. On the contrary other authors, as Zhang *et al.* [168] for instance, choose an ad-hoc time period length to compute the feature (*eg.* 5 days in the Zhang *et al.* study). Weng *et al.* notice that it takes approximatively 7 days to observe the 100-first tweets. Two types of information are considered to define features: the user-network topology, and the characteristics of the time-series that describe the temporal evolution of the topic. The user-network that is observed has 400 000 users and 10 millions links which represent a symmetric follow relationship, as studied by Romero *et al.*. In addition, communities are extracted from this user-network with the INFOMAP [139] and the LinkClustering algorithms [12]. The INFOMAP algorithm detects disjoint communities, whereas LinkClustering is an overlapping community detection method, as presented in Section 1.2.2. Features based on the user-network topology includes, but are not limited to: The number of communities in which the hashtag is used; The surface-size, that is to say the number of adopters of the hashtag plus their followers up to k steps away, the surface-size is also used by Zhang *et al.* [168]; The average step distance in the network of the n-first adopters. The time-series that describe the temporal evolution of the topic, are used to define several additional features. As for instance the "growth rate" that is the average time delta between two

tweets among the n-first tweets. These features are used in classical a machine learning framework, and evaluated against three baselines. The most advanced baseline relies on a linear regression. As in several other work, the prediction task is tested on the base-10 logarithm of the topic-activity measures. Authors conclude the models based on linear regression, which performs well in the user-generated-content-popularity prediction, are not usable to predict the topic-activity. In addition, authors note that their network-based approach outperforms the baselines for predicting the most viral hashtags (*ie.* the hashtags that were used in more than 10000 tweets) and the unused hashtags (*ie.* the hashtags that were adopted by 10 users at most). In addition a comprehensive analysis of the feature importance is provided.

In the previous studies, the topic-activity is predicted for the whole "lifetime" of the topic. That is to say, the topic-activity of a topic named z is equal to the number of user-generated-contents: $|\mathcal{C}_z|$ that are produced until the topic ceases to be active, or monitored. On the contrary, Ma *et al.* [108] or Tsur and Rappoport [151] propose to predict the activity for a one or several time-period(s). In this setting the activity of the topic named z during the time-period t is $|\mathcal{C}_z(t)|$.

Tsur and Rappoport [151] study the activity of a topic as a number of user-generated-contents. Their study is also based on Twitter, and the topic is equal to an hashtag. In this study, the time is discretized into time-periods that last one week. The topic-activity prediction is tackled as a regression problem. More precisely, for a topic named z, the quantity to be predicted is the base-10 logarithm of number of tweets that refers to z during several time-periods: $\sum_{i=0}^{k} |\mathcal{C}_z(t_i)|$, with k that varies in $\{10, 15, 20, 25\}$. This quantity is furthermore normalized in order to cope better with the variations of the total amount of tweet observed per time-period. Author discard hashtags that were already "popular" at the beginning of the observation, or were used less than 100 times. A "popular" hashtag named z is such that $|\mathcal{C}_z(t_1)|$ is greater than 10% of its highest activity for any time-period. The proposed features are based on three types of information: the user-network topology; the content of the topic; the characteristics of the time-series that describe the temporal evolution of the topic. Every feature is binary, therefore the continuous values are binned. The greatest attention is devoted to the features based on the content of the topic. These features includes, but are not limited to: The number of words in a hashtag (*eg.* #MusicMonday has two words); The length of the hashtag; The presence of lexical items which are matched against Wikipedia, and lists of holidays,

country names, celebrity names; The cognitive dimension such as positive or negative sentiments which are based on the LIWC project [147]; The features based on the user-network topology are straightforward, as for instance, the average and maximum number of followers. The features based on time-series that describe the temporal evolution of the topic reports the relative change of the topic-activity during the weeks 1, 2, 3 and 6 after that the topic became active. The results includes an analysis of the feature importance, and a comparison to a baseline which is a regression. As explained earlier, four forecast-horizons are considered: 10 weeks, 15 weeks, 20 weeks and 25 weeks. The regression based on the temporal features solely achieves the best results when one considers each feature sets separately and 10 or 15 weeks as forecast-horizon. On the contrary the regression based on the network solely achieves the best results when one consider each feature sets separately and 20 or 25 weeks as forecast-horizon. The "hybrid" regression, which relies on every feature, achieves the best results. The "hybrid" solution achieves a noticeable improvement with respect to the baseline. Indeed, the mean-squared-error of the "hybrid" solution is approximatively two times lower than the one obtained by the baseline. The computation of the features limits the use-cases for this solution. Indeed, one needs first to observe a fresh hashtag for six weeks, before being able to predict its activity.

Ma *et al.* [108] propose a study on the activity of a topic as a number of users that produce user-generated-contents that refer to this topic. In this study devoted to Twitter, a topic is represented by an hashtag. Their results are based those of the hashtags that were adopted by at least 25 users in one time-period. The forecast-horizon is equal to one time-period. A time period is equal to one day. This study outlines that the most effective feature is the number of users that already adopted the hashtag. Other features comparable to the ones proposed by Tsur and Rappoport are also tested.

All the previously presented studies relies straightforwardly on the machine learning framework presented in Section 1.2.5. However other approaches are proposed. For instance Guille *et al.* [74] adapt the independent cascade diffusion model which is presented in Section 1.2.3. Their model is first tuned to compute the probabilities of infection in the user-network. Afterwards, the forecast of the number of users that adopt a topic is obtained by a simulation of the diffusion for a topic of interest. The work of Yang and Leskovec [162] proposes another alternative method to predict the number of users that will adopt an hashtag. Indeed, they model directly the influence of each user that adopt

the hashtag of interest, on the diffusion rate of the hashtag.

Prediction of spikes of collective attention.

When the prediction is specifically devoted to spikes of collective attention, one considers time-periods that are noticeably shorter. Indeed, it is shown that social-media react in a fast-paced way as presented in Section 1.3.2. For instance Kwak *et al.* [94] observe that almost 31% of the trending topics last one day, and only 7% last more than 10 days. Although the prediction of spikes of collective attention are naturally based on the latest observations, they target a specific topic-activity evolution and can be distinguished from other studies that consider the latest observations for the topic-activity prediction.

The latest work of Kong *et al.* [92] aims at predict in real time the spikes of collective attention. The evaluation of this work is done in Twitter, therefore, as previously, a topic is considered to be represented by an hashtag. The time is discretized in time-periods that last one minute. This has to be compared with day and weeks, that are usually used other studies on the topic-activity prediction. Authors propose three sub-task for the prediction of spikes of collective attention. Firstly, a binary classification task that aims at decide if the hashtag will be the object of a spike of collective attention. Secondly, a regression task that aims at predict the number of time-period(s) (*ie.* minute(s)) before that the burst occurs. Thirdly, a regression task that aims at predict the number of time-periods that the spike will last. The features are based on three types of informations, as presented earlier. Some feature are inherited from the user-generated-content activity, as for instance the ratio of tweet that refers to the topic of interest and contains an URL, which was proposed by Suh *et al.* [145]. Features based on the user-network are considered. The user-network is built with respect to the user-mentions, as proposed by Romero *et al.*, and the re-tweet actions. The features based on the user-network includes the average degree, the density of the graph, and the order of the graph as presented in Section 1.2.1. Features based on the users behaviors are also considered as proposed by Zhang *et al.*. Authors also devoted a great deal of effort on the features based on times-series that describe the evolution of the topic. Author reports that these features are the most important ones for the prediction of the spikes of collective attention.

1.3.4 Predicting user-generated-content activity

The studies related to the user-generated-content activity problem are now presented. To this end, the definition previously presented in Section 1.4.1 are used. In these studies, the activity of a user-generated-content varies with respect to the social-media that is considered. For instance, in Twitter, the activity measure most frequently considered as the number of re-tweet that one tweet gets. Another activity measure in Twitter, is the number of replies that a tweet gets. In other social media, such as Vimeo or Youtube, the number of views is considered as the activity. In addition most of the social-media allows users to up-vote and down-vote each user-generated-content, the vote balance is also used as a activity measure. The activity of a user-generated-content has much more distinct definition than the topic-activity. It is important to stress out the difference between topic-activity and user-generated-content activity. Indeed to predict user-generated-content activity allows one to obtain a lower-bound of the topic-activity. The difference between these two tasks is widely acknowledged.

Most of the studies devoted to predict the user-generated-content activity are formulated in as supervised machine learning problem, as presented in Section 1.2.5. A *multi-class classification* might be considered, in such a case the activity value is binned. Another approach is to define the problem as a regression, in such a case the exact activity is predicted. The user-generated-content activity prediction is based on three types of information: (a) the characteristics of the user that produced the user-generated-content of interest, as for instance, the number of its followers; (b) the content of the user-generated-content, for instance the number of URL that it contains; (c) the characteristics of the time-series that represents the spread of the user-generated-content of interest. The majority of the propositions in this line of inquiry are tested in Twitter. Indeed in this social-media, almost all user-generated-content (*ie.* tweets) are publicly available. The most noticeable studies devoted to Twitter are proposed by: Petrovic *et al.* [127]; Naveed *et al.* [119]; Suh *et al.* [145]; Luo *et al.* [107] and Morchid *et al.* [116]. Other social-media are also studied, for instance Youtube, Vimeo, Digg or Reddit are studied by Szabo and Huberman [146], Figueiredo [55], Pinto *et al.* [129], and Lakkaraju *et al.* [97], respectively.

The prediction of user-generated-content activity drew a tremendous amount of work. However, this type of activity prediction does not directly to fulfill our industrial requirements. Therefore, rather than review each approach proposed for the user-generated-

content activity, we propose to review the study of Morchid *et al.* [116]. Indeed, this study is germane to topic-activity prediction because authors leverage topics that are discussed in the user-generated-content in order to predict the activity of the user-generated-content. In this study devoted to Twitter, the activity measure is the number of times a tweet gets re-tweeted. In order to define a *binary-classification* problem, this activity measure is compared to a threshold named θ. Hence, a tweet that gets more than θ re-tweets is a positive example, otherwise the tweet is a negative example. The threshold θ varies in from 10 to 90 with an increment of 10. The time is not discretized in this study, thus the re-tweets are counted as they appear before the end of the observations. The proposed approach is as follows. Firstly the keywords are extracted from the tweet for which activity has to be predicted. This task is a substantial burden. Indeed, as explained in Section 1.1, the user-generated-content are unconstrained and misspelling or new forms or spelling are frequently observed. Secondly, based on these extracted keywords, the features are computed. Some of these features relies on the activity of the topics that are associated to the keywords in the tweet. Thirdly, a classical supervised machine learning schema is applied, as presented in Section 1.2.5. The proposed approach is now presented in details.

Authors compare two methods for the extraction of the keywords in tweet. The first keyword-extraction method is based on the TF-IDF-RP measure as proposed by Salton [142]. This measure can be though of as a TF-IDF, where the score of a token named w is weighted by the number of token(s) in the tweet divided by the position of the first occurrence of w. In this keyword-extraction method the tokens that have the 10 highest TF-IDF-RP scores are selected as the keywords of the tweet. The second keyword-extraction method is based on the latent Dirichlet allocation as proposed by Blei *et al.* [25]. This generative topic-model defines topics as probability distributions over the tokens of a predefined vocabulary. The latent Dirichlet allocation, is applied on a corpus which has 1 billion tokens and includes Wikipedia and news articles. Authors define a semantic space of 5000 topics, each topic is truncated to keep only its 50 most likely tokens. In this keyword-extraction method the tokens of the tweet are scored with a measure that takes into account: (a) the similarity of a topic and the tweet in which the token are observed, and (b) the importance of the tokens within the topic of interest. The highest ranked tokens are selected to be the keywords of the tweet.

There are three features proposed in this study. The first one describe the activity of the tweet-keywords. This activity is based on several news feeds (*ie.* Real Simple Syndication feeds), observed before the creation of the tweet. The activity of a keyword

equals to its frequency in the whole news feeds weighted by its frequency in each news items which contains this keyword. The activity feature for a tweet, is the maximum of the activity for the tweet-keywords. The second feature is named "singularity". This feature captures the probability for the topics of a tweet to be associated. Solely, the two most likely topics are considered in this feature. In order to lower the computational cost of this features authors use a graph representation of the topical space and compare the topics with the symmetric Kullback-Liever divergence. The third feature stems from the *valence* measure. The *valence* of a token is based on the probability to observe a token in a positive context. In this study, a token observed near to a positive emoticon, such as instance "*:)*", is considered to be observed in a positive context. Rather than to differentiate positive and negative token, authors measure if a word is associated to a sensitive context. This measure is applied to each keyword of the tweet, the maximum of these scores defines the third feature.

Authors report results on a set of 4500 tweets. The train-set contains 90% of these tweets, and the test-set contains the remaining tweets. Authors compare the result of neural networks classifiers for the two keyword-extraction methods. The latent Dirichlet allocation allows to obtain the best performances. Authors also study the influence of the threshold θ, which is used to define the positive examples of the *binary-classification* task. For the majority of the threshold value the best predictor relies solely on the activity feature. The results of this predictor are stable when θ varies from 10 to 90. It is noteworthy that the combination of all the feature is generally outperformed by this single feature.

Conclusion

This chapter introduces the *social-media-mining* from a practitioner point of view, and presents extensively the prediction of the activity in *social-media*. In order to introduce the *social-media-mining*, the *social-media* are defined and the studies of their properties are presented. In addition, research questions that are related to *social-media-mining* are discussed. Moreover, practical questions such as data retrieval from social media, are addressed.

Ellison and Boyd [52] consider that social-media are web-based services that enable their users to: (a) have a uniquely identified profile, (b) produce, consume, and interact with stream of user-generated-contents, (c) connect publicly to other users. This defini-

tion emphasizes the interplay between user-generated-contents and the user-network that structures their diffusion among users. The user-networks are studied as graphs. As presented in Section 1.2.1, the characteristics of these graphs, although slightly varying from one social-media to another, are mostly comparable for any social-media. The typical user-network of a social-media has a node degree distribution that follows a power-law, with an average diameter lower than 6. This graph has tight clusters of low-degree nodes. These nodes correspond with regular peoples that are not broadly popular. The clusters of such nodes are linked together by high-degree nodes which correspond with broadly popular peoples.

Several questions that pertains to graph analysis applies to *social-media-mining*. Two of these questions are discussed in this chapter as they are closely related to the activity prediction. Firstly the community detection, secondly the information diffusion modeling. These two questions are now briefly summarized. The communities observed in the user-network have a role in the spread of user-generated-contents and thus in their activity. As presented in Section 1.2.2, there are several communities definition: disjoint, overlapping, and ego-based. For each of these definitions, several detection methods are proposed. The tremendous size of social-media makes necessary for these methods to be highly scalable. The modeling of information diffusion aims at describe how a user-generated-content flows through the user-network, it is presented in Section 1.2.3. There are several proposals to model the information diffusion. Most of these models are based on contagion models as the independent cascade model. The activity prediction can be considered as a byproduct of such models that describe how the information spread across the users of a network. The latest information diffusion models take into account the characteristics of each user, such as their acquaintance with the information to be spread.

As per Zafarani et al., the *social-media-mining* is the "process of representing, analyzing, and extracting actionable patterns from social media data". According to this definition, the *social-media-mining* bears from the machine learning. Indeed, as described in Section 1.2.5, the machine learning aims at extract knowledge from data, and the social-media are dedicated to the creation of data by their users. Hence, that user-generated data can be collected and automatically exploited with machine learning methods. As such, the prediction of the activity in social-media is a *social-media-mining* problem that can be tackled with machine learning. This general problem can be casted in more spe-

cific problems as the activity has several definitions. For instance, the activity might be considered per thematic (*eg.* a brand, a product, a news item) or for a single user-generated-content. Moreover, the forecast horizon of the activity prediction varies from hours to months. Hence, the activity prediction might target burst of activity as well as seasonal trend. These various settings drew a substantial amount of work. The studies dedicated to the activity prediction are presented in Section 1.3, and summarized below.

The most well known activity prediction setting is the prediction of the spikes of collective attention presented in Section 1.3.2. This setting is often studied with data collected on Twitter. Indeed, the majority of the user-generated-contents are public in Twitter, and the spikes of collective attention (*ie.* the most popular topics) are reported in real time. The studies of spikes of collective attention concluded in the existence of few types of spikes. These spikes can be categorized with respect to the type of event that triggered them, and to the distribution of their activity through time. These spikes are mostly supported by user-generated-content that are produced independently (*ie.* without referring explicitly any other user-generated-content). For this setting, the forecast horizon is narrow. Another prediction setting is to predict the activity of a single user-generated-content as presented in Section 1.3.4. Then, only the user-generated-contents that refer explicitly (*eg.* to re-tweet, to like, to share, etc.) to the user-generated-content of interest are considered in the activity measurement. A third setting is to predict the activity for a whole thematic, as for instance "smartphones". A thematic groups a great quantity of user-generated-contents that are not explicitly related. Section 1.3.3 presents studies dedicated to this setting. Most of these studies bears on the machine learning framework, and make use of features that describe: (a) the content of the topic, (b) the network of active user for the topic, and (c) the variations through time of the topic activity. Most of these studies are made with respect to one single *ad-hoc* social-media and feature set. Some of the proposed approaches might not be suitable for actual day to day prediction. For instance, to compute the average step-distance in the adopters-network requires an up to date knowledge of the user-network which is not practical with respect to the constraints presented in Section 3.1.

To forecast the activity of a thematic, as a routine industrialized task, hasn't been studied yet, although, it requires to match specific constraints. The remainder of this dissertation is dedicated to study of the industrialization of the activity prediction. To this end, a generic framework that describes most of the social-media is defined. A set of

easily computable features is proposed. A scalable data collection system is implemented. Numerous experiences are done to validate the results of this approach, and to test the activity prediction as a routine industrialized task. From now on, the prediction of the topic activity is referred to as the "activity prediction problem".

Chapter 2

Generic framework for activity prediction

Contents

Introduction

This chapter presents a generic framework that allows to define a *social-media-mining* problem independently of an actual social-media. Three examples are provided to illustrate how the generic framework allows to describe actual social-media. These examples cover two popular social-media: Twitter and Facebook, and the widely used, yet less trendy, bulletin-board-system. That framework is used to study the activity prediction problem under practical constraints. These constraints aims at be able to predict the activity in a daily industrialized routine.

The first section of this chapter presents the generic framework. This framework relies solely on public information. Indeed, privileged access to non public data is usually impossible or to expensive fees and administrative burden. The generic framework describe a social-media such that the user-generated-contents carry textual information and the interactions among users entail for involved user-generated-contents to be explicitly grouped together.

The second section of this chapter defines the activity prediction problem with the generic framework. The activity prediction is formalized independently of an actual social-media. More precisely three different definition proposals are made. Each of these proposition aims a particular applicative scenario. The first definition is suitable for one whose aims at predict the activity of topics that exhibit brief cycles of activity burst and decrease. The second definition is suitable for one that aims at predict the activity of topics with seasonal activity patterns. The third definition is suitable for one that aims at predict the relative importance of each topic among a group of topic.

The last section of this chapter presents the features used to predict the activity in a supervised machine learning setting. These features are defined within the generic framework. Therefore the feature are easily adapted to various social-media. These features are divided in two groups. The first group of feature is defined for a single topic. The second group of feature is defined for a pair of topic and aims at measure interaction between topics.

2.1 Generic framework

The generic framework allows to describe several social media. Hence, one can use the generic framework to define a *social-media-mining* problem, and study this problem with data collected in several different social-media. To apply this framework to a social-media is straightforward. Roughly, the framework can be used to describe any social-media where user-generated-contents come along with text. More precisely, this framework might be used on any social-media that fulfills the following requirements:

1. the user-generated-contents have to be: (a) uniquely identified; (b) provided with a unique user identifier and a time stamp; (c) associated to a textual content;

2. two user-generated messages, that result from an interaction among users, have to be explicitly grouped. For instance, a pair of messages that contains a question form a user and the answer provided by another user, is the result of an interaction among users, and must be identifiable as such.

As mentioned in Section 1.2.1, the relations among users of a social-media define a user-network. This network provides plenty informations, for instance it is used to extract communities or to model information spread, as presented in Sections 1.2.2 and 1.2.3, respectively. However, the generic framework does not describe the user-network. Indeed to capture and keep up-to-date such networks is not practical for most social-media. Consider for instance Twitter, it has a user-network that counts 225 millions active users [80]. Therefore, to retrieve the whole user-network requires to bypass the Twitter application programming interface. In order to get a complete snapshot of the user-network of Twitter, Gabielkov [60] had to use a distributed system. It took four months for this distributed system to capture the whole user-network of Twitter. One might then consider to sample the user-network, however it requires to be able to control the sampling bias, a non trivial task, as presented in Section 1.2.4. Moreover, even with a ready to use snapshot of the user-network, it might be necessary to take into account the evolutions of the user-network (*eg.* a user of Facebook might add and removes friends). To keep up-to-date a snapshot of the user-network remains a costly burden. In addition to these concerns about the data collection, the meaning of user network vary significantly from a social-media to another. For instance, in Facebook, the predominant relation is symmetric and denotes the friendship. Conversely, in Twitter, the predominant relation is asymmetric and denotes the interest of one user to another. Therefore, in order to

match our industrial requirements, the generic framework does not take into account the user network available in social-media.

2.1.1 Definition

The generic framework has five components:

(a) The set of messages, that are exchanged in the social media, referred to as $\mathcal{M}$;

(b) The set of users, that exchange messages in the social media, referred to as $\mathcal{U}$;

(c) The set of topics, which vary with respect to the aggregation function, referred to as $\mathcal{Z}$. The aggregation function is defined as $\nabla : \mathcal{M} \mapsto \mathcal{Z}$, it maps a textual content to one or several topics in $\mathcal{Z}$;

(d) The set of time-stamps, which are observed within the social media, referred to as $\mathcal{S}$. The time resolution varies form one social-media to another. For instance, in Twitter, each tweet has a time stamp that is precise up to the second;

(e) The set of time-periods, which allow vary the time resolution of the time stamp, referred to as $\mathcal{T}$. The time resolution of a time-period is fixed prior to the study. The time-periods allows one to study several social-media in a unified time resolution. The time-periods are defined as a surjective function from the time stamp to the time-periods $\Phi : \mathcal{S} \mapsto \mathcal{T}$.

The generic framework has two entities: the `content`, it models the user-generated-content with a text, and the `discussion`, it models groups of user-generated messages. For instance a `discussion` could be a question and each subsequent answer it got. The ways to group several `contents` vary from a social media to another. Therefore the semantic of a `discussion` varies, but it constantly represent groups of content which are created as users interact.

Definition 7 `content`. *A user $u \in \mathcal{U}$, that generates a message $m \in \mathcal{M}$, at time $s \in \mathcal{S}$, is modeled in the generic framework as a* `content` *which is a quadruplet:*

$$\langle m, \nabla(m), u, \Phi(s) \rangle \in \mathcal{M} \times P(\mathcal{Z}) \times \mathcal{U} \times \mathcal{T}$$

The set of topics associated to the message, named $\nabla(m)$, varies with respect to ∇ the aggregation function. The set of any existing `contents` *is referred to as $\mathcal{C}$. $\mathcal{P}(\mathcal{Z})$ designates the partition the set of topics $\mathcal{Z}$.*

The set of discussions is referred to as $\mathcal{D}$, a `discussion` $d_i \in \mathcal{D}$ is defined as a sequence of temporally ordered `contents`. This sequence may evolve according to the user interactions thus the definition of a `discussion` takes into account the time of the observation.

Definition 8 `discussion`. *A* `discussion` d_i *observed at time-period* τ *has* $|d_{i,\tau}|$ `contents` *and is defined as follows:*

$$d_{i,\tau} = \left\{ \langle m_i^1, z_i^1, u_i^1, t^1 \rangle \ \ldots \ \langle m_i^{|d_{i,\tau}|}, z_i^{|d_{i,\tau}|}, u_i^{|d_{i,\tau}|}, t^{|d_{i,\tau}|} \rangle \right\} \in \mathcal{C}^{|d_{i,\tau}|}$$
$$\text{with} \quad t^{|d_{i,\tau}|} \leq \tau$$

Consequently, in the discussion d_i*, the* j^{th} *message is* m_i^j *from the author* u_i^j *at time-period* t^j*; the topics associated to this message are* z_i^j*.*

Definition 9 *(*`discuss` *function) This function groups* `contents` *into* `discussions`*, its domain is as follows,* $\texttt{discuss} : \mathcal{C} \times \mathcal{C} \mapsto \mathcal{D}$*, and its definition is:*

$$\texttt{discuss}(c_i, c_j) = \begin{cases} d_k \cup c_j & \text{if } \exists\, d_k \in \mathcal{D} :\ c_i \ \in \ d_k \\ \\ \{c_i, c_j\} & \text{else.} \end{cases}$$

Definition 10 *(*`users`*,* `messages`*,* `topics`*, and* `activity` *functions) These four functions provide informations about the* `contents` *and* `discussions`*. These informations are used to defined the features presented in Sections 2.3.1 and 2.3.2. In these four definitions,* $\mathcal{P}(X)$ *designates the partition of a set* X*.*

1. $\texttt{users} : \mathcal{Z} \times \mathcal{T} \mapsto \mathcal{P}(\mathcal{U})$ *that provides the subset of users using a topic* z *at time-period* t*:* $\texttt{users}(z,t) = \{u \mid \langle m, \hat{z}, u, t \rangle \in \mathcal{C} \wedge z \in \hat{z}\}$*. We abbreviate it with* $\mathcal{U}_{t,z}$*;*

2. $\texttt{messages} : \mathcal{D} \mapsto \mathcal{M}^m$ *that yields a set of message exchanged within* $d_{i,t}$*:* $\texttt{messages}(d_{i,t}) = \{m \in \mathcal{M} \mid \exists \langle m, z, a, t' \rangle \in d_{i,t}\}$*;*

3. $\texttt{topics} : \mathcal{D} \mapsto \mathcal{Z}^m$ *that yields a set of topics used within* $d_{i,t}$*:* $\texttt{topics}(d_{i,t}) = \{z \in \mathcal{Z} \mid \exists \langle m, z, a, t' \rangle \in d_{i,t}\}$*;*

4. $\texttt{activity} : \mathcal{Z} \times \mathcal{T} \mapsto \mathbb{N}^+$ *that provides the activity observed for a topic* z *at the time-period* t*:* $\texttt{activity}(z,t) = |\{\langle m, \hat{z}, u, t \rangle \in \mathcal{C} \mid z \in \hat{z}\}|$*.*

During a time-period the activity of a topic is the number of `contents` related to this topic that are published. However, the activity of a topic during a time-period might be defined in several other ways. Consider a probabilistic aggregation function which maps a message to a distribution of topics. In such a case, the activity function would be weighted with respect to the probability distribution.

The generic framework can describe various social-media. Three different social-media are presented below: Twitter, Facebook, and the bulletin board systems. We show how to describe each of these social-media with the generic framework.

2.1.2 Illustration with Twitter

In this social media the communications are performed using size-bounded text messages called "tweets". Such messages can be the subject of a reply, or a "re-tweet", the latter amounts to repeat the tweet and to quote the original author. In this social media, users have a "timeline" that list their tweets. Users can furthermore "follow" each others to be notified of every new tweets from a followed user.

Twitter can be described in the generic framework. As presented below, the entities and functions of the generic framework allows to described the way Twitter works. The `content` entity corresponds to a tweet and a `discussion` is to a sequence of tweets that are related to each others. Such a sequence is obtained in twitter when users re-tweet or reply to existing tweets. Therefore the `discuss` function is mapped to both the reply and the re-tweet actions, indistinctly. As a consequence, a `discussion` is a temporally ordered sequence of tweets, in which any tweets, except the oldest one, is either a reply or a re-tweet of a previous tweet that belongs to this `discussion`. Finally the functions `users`, `messages`, `topics` and `activity` don't need to be defined, as they are defined with respect to the *content* and *discussion* entities. Figure 2.1 illustrate how the generic framework describes Twiter. It might exist tweets without any textual information. These tweets are simply ignored, indeed they cannot be associtated to any topic.

2.1.3 Illustration with Facebook

Facebook offers several privacy settings and allows to share informations of various type, for instance, text, image, video, event, application, and so on. In the remainder, solely

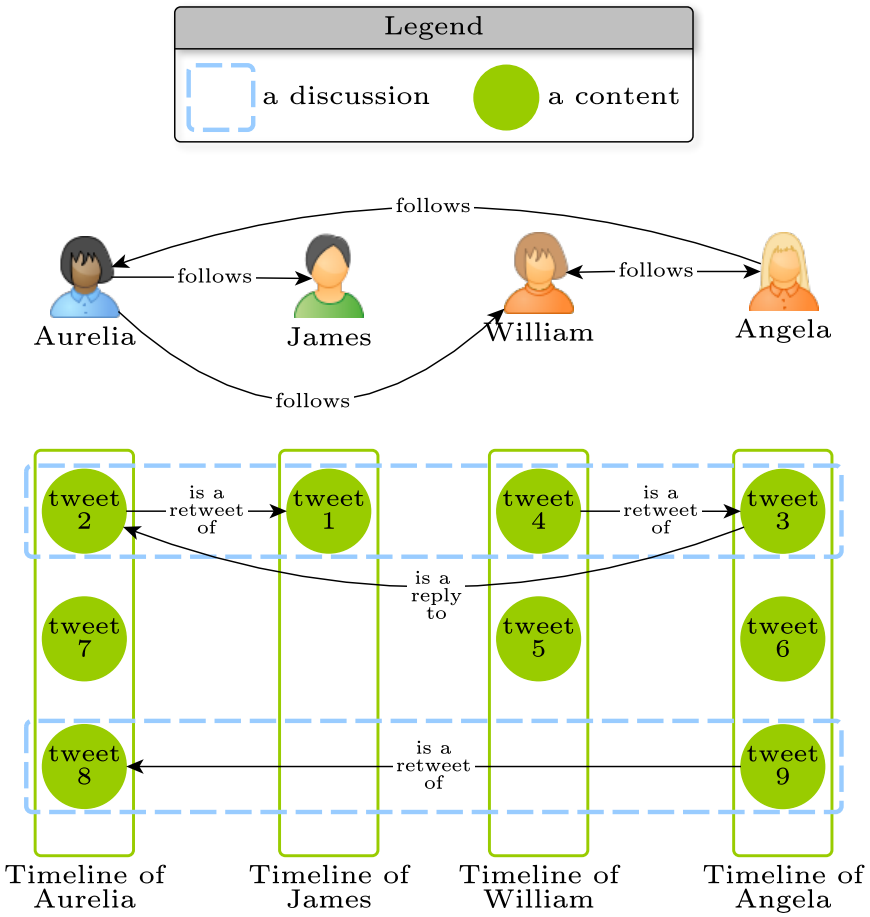

Figure 2.1: Twitter described with the generic framework. The top part of this illustration presents the tweets grouped with respect to the timeline of each user, and the corresponding entities from the generic framework. The first line of this figure presents an interaction involving four users. This example is as follows: James published a "*map of beer consumption in USA*"; Then Aurelia shared this visualization with the users who follow her. Afterward Angela replied to Aurelia with a "*map of vine consumption in France*". Finally William shared the vine map to users not represented in this illustration. The lower part of this illustration describes the network of users that is not captured by the generic framework.

the retrievable publications associated to at least one text message, are considered (*eg.* an image that has a caption). In order consider user-generated-contents without textual information (*eg.* images, videos, and so on), one has to propose specific methods. For instance one could propose an image classifier that labels images with respect to the items that are recognizable in the image, a non trivial burden. In Facebook, each publication

can be discussed. In such a case, comments, that are themselves publications, can be discussed as well. The "like button" is another way to interact with a content. This button allows to declare one's interest for a content. The use of the "like button" implies not any content creation. Howecer, the user of the "like button" declares explicity his-or-her interest for the liked content. The usage of the "like button" is similar to the re-tweet in Twitter. Therefore, the "like button" is mapped in the generic framework, to a `content`. This `content` is produced by the user of the "like button", and has the same textual information than the original content that were "liked".

The generic framework can describe Facebook in the same way that it describes Twitter. Indeed, the `content` entity represents any publication that has one text message or more. As such, the generic framework does not take into the type of the user-generated-content that is describe with a `content`. The `discuss` function describes the actions used to comment or like any existing content. Hence a `discussion` groups a publication along with any comment(s) it got. Like in Twitter, the functions `users`, `messages`, `topics` and `activity` don't need to be defined. Indeed, these functions are defined with respect to the *content* and *discussion* entities.

In addition to Facebook and Twitter, the generic framework describes the "bulletin board systems" also known as "message board" or "forums". The recent growth of numerous social-media outshines the forums, yet the forums remains widespread, and worthy of study.

2.1.4 Illustration with a message board

In a message board the users publish messages that are named "posts". Each post is organized as it would be by pinning it on cork pinboard. In a message bord, a sequence of related messages is named a "thread". A message board has multiple boards to publish in. Each board is dedicated to a specific topic (*eg.* smartphones, laptops, holidays). It is mendatory for a user to choose the right board to publish his-or-her message. The user chooses the board in accordance with the topic of the message he-or-she wants to publish. This collaborative organization allows a reader to explore the contents in accordance of a topic. In other social media, as for instance Facebook, the contents aren't grouped with respect to their topic, as presented in Section 2.1.3 and defined in Section 1.1.

The generic framework describes a message board. Indeed, the `content` describes a post and the `discussion` describes a thread. Hence the `discuss` function describes the reply action, as illustrated in Figure 2.2. In this social media it might exist posts without any textual information, these one are ignored, as in Facebook or Twitter. Like in Facebook the functions `users`, `messages`, `topics` and `activity` don't need to be defined, as they are defined with respect to the *content* and *discussion* entities.

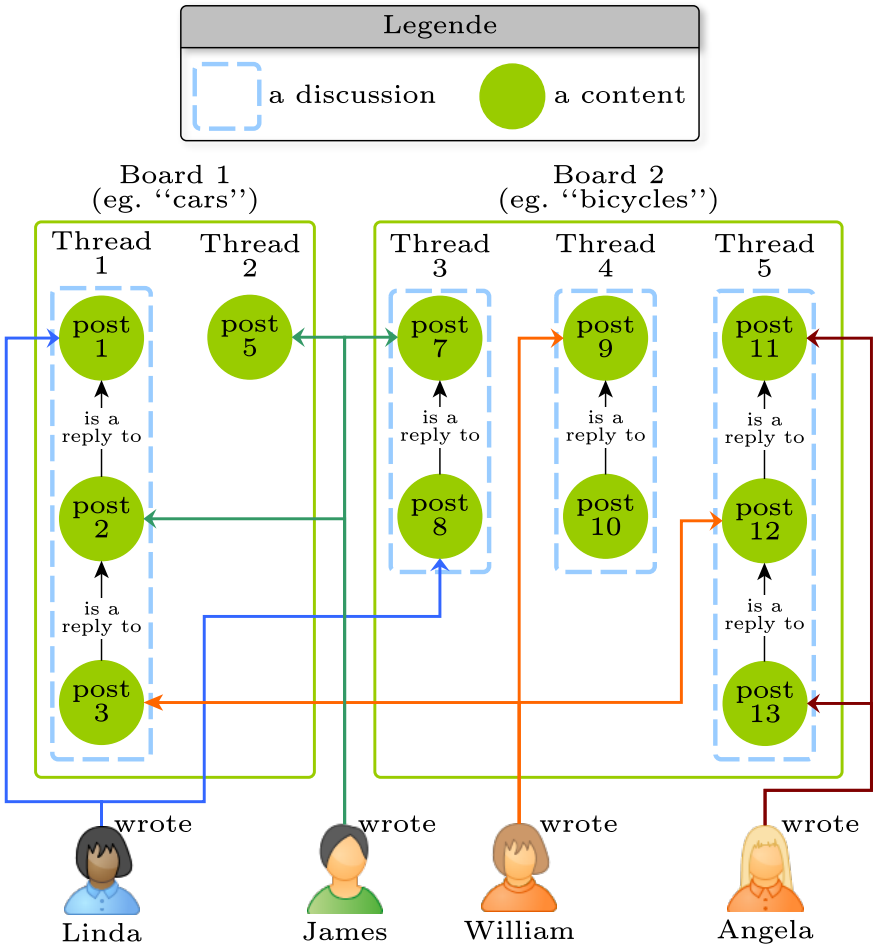

Figure 2.2: A bulletin board system described with the generic framework. The top part of this illustration presents a bulletin board system with two boards. The corresponding entities of the generic framework are also presented in the top part of the illustration. In the first board, the posts are related to cars. The second board contains posts about bicycles. A post example from the second board is "*which is the best bicycle for commuting by a rainy day?*". The lower part of this illustration shows which is the author for each post in the bulletin board system.

2.1.5 Aggregation of the user generated contents

The generic framework defined in Section 2.1.1 is usable for the study of several *social-media-mining* problems. To predict the activity it is necessary to aggregate the user-generated-contents into topics, or domains of interest. In order to illustrate the purpose of this aggregation consider these tweets: "*The new samsung galaxy s5 is out, I need one*"; "*Galaxy s5 is waterproof, it's awesome*" and "*Damn! the xperia Z2 is over the top*". These tweets belongs to the "high-end smartphones" topics because "galaxy s5" and "xperia Z2" are high-end smartphone models from samsung and sony respectively. Nonetheless these tweets could have been grouped differently and topics could be "samsung" and "sony". To aggregate user-generated-contents into topics is not a trivial task. Indeed new spellings, and domain specific expressions, and polysemy are usual in user-generated-contents.

Formally this aggregation of user-generated-contents is a dimensionality reduction from an high dimensional space defined by the words used in the user-generated-contents to lower-dimensional space defined by topics. In the lower dimensional step groups of words are interpreted as topics. This aggregation step is often referred to as a topic model. Numerous topic models were proposed, some of them are specifically designed to be used in social-media. One can use the generic framework with an arbitrary topic-model as long as this topic model relies solely on the text carried within the user-generated-contents. For instance the latent Dirichlet allocation proposed by Blei *et al.* [25] can be used in generic framework. On the contrary, one can not use a topic-model that is based on the location of the users such as the one proposed per Hong *et al.* [79]. Two families of topic models are now briefly presented.

The latent semantic analysis (LSA), also known as latent semantic indexing, is proposed per Deerwester *et al.* [59]. The latent semantic analysis is a dimensionality reduction method designed for textual content. However the dimensions of the lower dimensional space are not directly interpretable as topics. The LSA, just as most topic models, is based on the bag-of-word representation. Hence the order of words in a user-generated-content is discarded. With this representation, the corpus of user-generated-contents is described by a word-document-matrix named $\mathbf{X}$. The scalar $\mathbf{X}_{i,j}$ describes the usage of the i^{th} token of the vocabulary in the j^{th} user-generated-content. The LSA uses singular values and singular vectors of the word-document-matrix to reduce it. The singular val-

ues of $\mathbf{X}$ can be computed by the mean of the singular-value decomposition, this matrix factorization method splits the word-document matrix in three matrices: $\mathbf{X} = U\Sigma V^T$. Where the matrix U describes the words with respect to the left singular vectors. The matrix V describes the documents with respect to the right singular vectors. Finally Σ is a diagonal matrix that contains the singular values associated to the left and right singular vectors. A complete description of the singular-value decomposition is proposed in [69]. The LSA makes use of this decomposition to approximate $\mathbf{X}$ in a lower dimensional space. This problem is referred to as the low-rank approximation problem. It has been proven that this problem admits a solution when the quality of the approximation is measured with the Frobenius norm [51]. This solution is as follows.

$$\hat{\mathbf{X}} = \hat{U}\Sigma_k\hat{V}^T$$

Where Σ_k is the approximation of Σ that contains only the k largest singular values from Σ, and $\hat{U}$ and $\hat{K}$ contain the corresponding singular vectors from U and V. The LSA is based on the approximation $\hat{\mathbf{X}}$ which describes each user-generated-content as a vector of dimensionality k. On the contrary, the latent Dirichlet allocation is a generative probabilistic topic model proposed by Blei *et al.* [25]. A topic corresponds to a probability distribution of word occurrence. This topic model is also based on the "bags-of-words" assumption. The generative process of LDA is controlled by two parameters: θ that is a per-document topic proportions, and β that is a per-corpus topic distribution. The generative process is used for each an any document of the corpus, it can be summarized as follows:

1. From Poisson law, choose the document length named N_d;
2. From the Dirichlet distribution, choose the multinomial topic distribution named θ_d that is specific to the current document named d;
3. Repeat N_d times:
   (a) Choose a topic named z_n from Multinomial(θ_d)
   (b) Choose a word from $p(w_i|z_n, \beta)$ a multinomial probability conditioned on the topic z_n

The observations are here documents $d = (w_1, \ldots, w_{N_d})$ as per the *bag-of-words* assumption. Inversely: per-word topic assignment, per-document topic proportions, and per-

corpus topic distribution are latent variables. The probability associated to an observation is as follows:

$$p(d) = p(\theta|\alpha) \prod_{n=1}^{N_d} p(z_n|\theta_d) * p(w_n|z_n, \beta)$$

This equation reflects the generative process. First a topic distribution is drawn from the Dirichlet distribution ($p(\theta|\alpha)$), then according to this distribution, for each word-token of the document, a topic is chosen $p(z_n|\theta)$. Having chosen the topic (z_n) associated to the n^{th} word-token the β distribution (which maps topics to words) can be used to draw the n^{th} word using $p(z_n|\theta_d) * p(w_n|z_n, \beta)$. To learn topics from a corpus amounts to maximize the log-likelihood of topic model to produce the observed documents. Since the quantity $p(d_i|\alpha, \beta)$ cannot be computed tractably, multiples approximations methods have been proposed. Such methods are for instance: the collapsed Gibbs sampling as proposed by Porteous *et al.* [130]; the collapsed variational inference as proposed by Teh *et al.* [148]; the expectation propagation as proposed by Minka and Lafferty [113]. In order to learn topics from a corpus with LDA, one has first to set the number of topics. Ramage *et al.* [135] applied this family of topic model to user-generated-contents from Twitter.

The aggregation of user-generated-contents depends on the envisaged application. In our industrial context the aggregation is straightforward, it groups user-generated-contents with respect to a list of products names. With this aggregation method, all the user-generated-contents that mention the same product-name are grouped in one topic that is named after the product. These product names are made of one or several tokens as for instance "*Nvidia GeForce*" which refers to a nvidia product line.

This section introduced the generic framework, and presented how it can be used to describe Twitter, Facebook, and the messages boards. The generic framework relies solely on public data. The generic framework is used to study *social-media-mining* problems such as the activity prediction that is presented in the next section.

2.2 Activity prediction problems

This section presents three formal definitions of the activity prediction problem. Each of these definition is tailored for specific applicative requirements. As presented in Section 1.3 the activity prediction problem drew a considerable amount of work. The studies presented in Section 1.3 are generally devoted to one particular applicative need, as for

instance: activity burst prediction, or seasonal trend prediction. Without being exhaustive, the objective of this section is to cover a broad share of the possible applications of the activity prediction problem. To this end three definitions of the activity prediction problem are proposed and the generic framework is used so that the proposed definitions are not tie with a particular social-media. The first definition proposal is the magnitude prediction, it aims at predict the activity volume change on a seasonal basis. The second definition proposal is the buzz classification, it aims at predict which topics will undergo an activity burst in the near future. The third definition is the rank prediction, it aims at predict the relative importance of each topic from a group of topics.

2.2.1 Magnitude prediction

The magnitude prediction problem is defined with respect to the generic framework. For each topic z in the set of topics $\mathcal{Z}$ at each time-period t, we observe an m-dimensional vector $X(z,t)$ the features of which are described in Section 2.3. For the time interval $[t-\alpha;t]$, these observations are summarized into an m-variate time series $\boldsymbol{X}(z,[t-\alpha;t])$ and the problem we face is the one of predicting the value of a target variable $Y(z,]t;t+\delta])$. Y is a univariate time series that indicates the activity of z during the time interval $]t;t+\delta]$. The activity of z can be defined in several ways. For instance, one may consider the feature named ACT and defined in Section 2.3.1. In many practical situations, the values of the variable Y can be observed during the time interval $[t-\alpha;t]$, then it corresponds to one dimension of the vectors X and one wants to predict its future values knowing its past values and the past values of the other $(m-1)$ variables. The above problem corresponds to a regression problem as presented in Section 1.2.5. The goal of this regression is to find a function f, in a family $\mathcal{F}$, that relates the target variable to the observed ones with respect to $\boldsymbol{\Omega}$ the set of parameters used by f.

$$Y(z,]t,t+\delta]) \approx f(\boldsymbol{X}(z,[t-\alpha;t]),\boldsymbol{\Omega}) \tag{2.1}$$

In equation 2.1, the value of δ controls the forecast horizon and the value of α controls the history size. Defined as such, the activity prediction problem is used to forecast seasonal activity growth and decay. Several studies are devoted to this formulation of the activity problem, as for instance the one from Tsur and Rappoport [151] or Ma *et al.* [108].

2.2.2 Buzz classification

In the buzz classification one does not want to predict the actual activity values, that is to say $Y(z,]t,t+\delta])$. Instead, the objective is to predict if a topic z will have a burst of activity in the near future. With buzz classification, the values of Y are mapped to $\{-1,+1\}$ by a labeling function. Hence, this problem is a binary classification problem, as presented in Section 1.2.5. Three labeling functions are proposed to match distinct applications, as described below.

$$\texttt{relative\_label}(z) = \begin{cases} +1 & \text{if} \quad \dfrac{\mu(Y(z,]t;t+\delta]))}{\mu(Y(z,[t-\alpha;t]))} \geq \sigma \\ -1 & \text{else} \end{cases} \tag{2.2}$$

In this labeling function μ stands for the mean, and the threshold σ controls the minimum level of burst that is deemed valuable. For instance, a topic whose activity double is labeled positive when then threshold is $\sigma = 2$. More precisely, the activity of the topic during the time-periods that span from t to $t+\delta$ have to be at least the double of the activity previously observed during the time-periods that span from t_0 to t. A second labeling function is proposed, indeed specific applications, as those that imply costly actions as human processing, may require an absolute threshold. In this case a threshold σ is fixed, and a keyword is labeled positive when its activity is above the threshold σ. This labeling function is described below.

$$\texttt{absolute\_label}(z) = \begin{cases} +1 & \text{if} \quad \mu(Y(z,]t;t+\delta])) > \ \sigma \\ -1 & \text{else} \end{cases} \tag{2.3}$$

The threshold named σ is identical for every keyword. Hence, an highly active topic that stays active, and a topic that goes from feebly active to highly active might be labeled identically. With this labeling function, the most active topics are likely to produce positive instances only. On the contrary the less active topics are likely to produce negative instances only. In order to obtain efficient models this labeling should be used with topics of comparable activity. Several studies are devoted to this formulation of the activity problem, as for instance the one from Romero *et al.* [137]. Finally, if one is interested in the absolute activity-growth the `absolute_label` function is updated as follows:

$$\texttt{growth\_label}(z) = \begin{cases} +1 & \text{if} \quad \mu(Y(z,]t;t+\delta])) - \mu(Y(z,[t-\alpha;t])) > \ \sigma \\ -1 & \text{else} \end{cases} \tag{2.4}$$

With this function, the positive examples are topics which activity between t and $t+\delta$ increased of σ or more, with respect to their past activity between $t-\alpha$ and t.

2.2.3 Rank prediction

In the rank prediction the activities of several topics are considered. The objective is to learn a ranking function f that arrange a group of topics with respect to theirs upcoming levels of activities. The ranking function function is defined with respect to the $X(z,t)$ the m-dimensional features vector that represents the topic z at each time-period t. In order to simply the notation, the m-dimensional features matrix that represents the whole set of topics $\mathcal{Z}$ at each time-period from t to t' is referred to as $X(\mathcal{Z},[t,t'])$. The ranking function is defined as follows.

$$f : X(\mathcal{Z},[t-\alpha;t]) \rightarrow R(\mathcal{Z},t+\delta) \tag{2.5}$$

In this ranking function $R(\mathcal{Z},t+\delta)$ is a ranking over the topics in $\mathcal{Z}$ for the time interval $[t;t+\delta]$. The ranking function f is learned on a training set consisting of vectors representing topics during the time interval $[t-\alpha;t]$ and associated with their activity value in the time interval $[t;t+\delta]$. Learning to rank approaches have been applied to social media in order to enhance user experience by providing the most relevant user-generated-contents to each user. For instance Duan *et al.* [48] or Uysal and Croft [152] ranks tweets with respect to user interest or ongoing events. We address here a different problem that is to learn to rank keyword with respect to their upcoming activities. The learning to rank methods can be divided into three main categories: pointwise, listwise and pairwise approaches.

With the pointwise approaches one assumes that each topic has an ordinal ranking score. Then, ranking is then formulated as a regression problem on this ordinal score. In this situation, the ordinal score of a topic is its activity on the time-period at the forecast horizon. Hence consider listwise approach amounts to solve the magnitude prediction for each topic to be ranked. According to Chapelle *et al.* [35], Pointwise approaches do not consider the interdependency among topics.

In listwise approaches a train instance is made of the whole ranking of every topics for a given time-period. Listwise techniques aim to directly optimize a ranking measure, so they generally face a major problem of dealing with non-convex, non-differentiable and discontinuous functions. Some approaches have been proposed to solve this problem by using convex surrogate functions for ranking objectives. The interested readers are referred to the work of Valizadegan *et al.* [153].

In pairwise approaches, the ranked list is decomposed into a set of document pairs. In this setting, ranking amounts to train a binary classifier. The inputs of this classifier are pairs of topics $(z, z') \in \mathcal{Z}^2$. The labeling function is as follows $label((z, z')) \Leftrightarrow rank(z) \geq rank(z')$. SVM is undoubtedly one of the most popular classifiers used to perform binary classification on the pairs of documents for ranking [82]. Other adaptation of popular classifiers to pairwise ranking, like RankBoost which minimizes the exponential loss over document pairs [58], has also attracted attention in the recent past. More recently, some work considered a smooth approximation to the gradient of the ranking loss instead of searching for a smooth and convex approximation to the ranking loss itself as proposed by Burges [34], McAllester *et al.* [111] considered a direct optimization of the ranking loss function. The interested readers are referred to the work of Cohen *et al.*, Freund *et al.*, and Thorsten, [41, 58, 82] for an overview of the pairwise approach.

These three activity prediction problems cover numerous applicative scenario. In addition, these definitions relies simply on the definition of topics, time-periods, and activity measure. Therefore, one can consider the activity prediction for various social-media and applicative needs. The next section introduces features defined with the generic framework in order to tackle these three activity prediction problems.

2.3 Features for activity prediction

This section defines the features that are proposed to tackle the activity prediction problems in a supervised machine learning setting. These feature are defined within the generic framework, as proposed in Section 2.1.1. Therefore the proposed features are defined once, but can be easily adapted to most of the social-media. The features are defined with respect to the entities of the generic framework. These entities are: the set of contributions named $\mathcal{C}$; the set of users named $\mathcal{U}$; the set of discussions named $\mathcal{D}$; the set of topics named $\mathcal{Z}$; and the set of time-periods named $\mathcal{P}$. There are two kinds of feature:

(a) The features that are defined for one topic and at least one time-period. They do not describe correlations between topics. These features are presented in Section 2.3.1;

(b) The features that are defined for a couple of topics and at least one time-period. They are used to capture interaction between topics. These features are defined in Section 2.3.2

2.3.1 Single topic features

1. **Activity** (ACT). This feature gives the quantity of `content` produced for a topic z at a time-period t. The definition of this feature, noted $\text{ACT}(t, z)$, is equal to the `activity` function defined in Section 2.1.1 – Definition 10. Therefore this feature is defined as follows.

$$\text{ACT}(t, z) = |\{\langle m, \hat{z}, u, t\rangle \in \mathcal{C} \mid z \in \hat{z}\}|$$

2. **Number of Active Discussions** (NAD). This features gives the quantity of `discussions` that is active during the time-period t, and has at least one user-generated-content that match the topic z. The definition of this feature noted $\text{NAD}(t, z)$, is as follows.

$$\text{NAD}(t, z) = |\{d_t \in \mathcal{D}_{t,z} \mid \exists\langle z, a, \tau\rangle \in d_t \wedge \tau = t\}|$$

3. **Number of New Discussions** (NND). This feature gives the number of `discussions` that is created during time-period t and that has at least one user-generated-content that match the topic z. This feature is referred to as NND and defined as follows.

$$\text{NCD}(t, z) = |\mathcal{D}_{t,z} \setminus \mathcal{D}_{t-1,z}|\,;$$

4. **Number of Users** (NU). This feature describes the number of users, that have produced at least one `content`, at the time-period t, for a topic z. This feature, referred to as $\text{NU}(t, z)$, is defined as follows.

$$\text{NU}(t, z) = |\mathcal{U}_{t,z}|$$

   This set is computed by the function `users`, presented in Section 2.1.1;

5. **User Engagement Level** (UEL). This feature gives the number of users which had never discussed a topic z before the time-period t, and started discussing it at t. In order to compute that feature for a topic z, and time-period t, one uses all the `content` captured for the topic z until the time period t. One furthermore uses the set-theoretic difference of $\mathcal{U}_{t',z}$ and $\mathcal{U}_{t'-1,z}$ with t' that varies from 1 to t. Therefore the User Engagement Level, that is noted $\text{UEL}(t, z)$, is defined as follows.

$$\text{UEL}(t, z) = \left|\mathcal{U}_{t,z} \setminus \bigcup_{i=0}^{t-1} \mathcal{U}_{t,z}\right|$$

   The value of user engagement increases when a topic captures numerous users which weren't interested in this topic before.

6. **Normalized Activity** (NACT). This feature copes with singular events that induce variations of activity for numerous topics at once. Such events modify the sum of activities observed at a time-period t. Hence this feature, noted $\text{NACT}(t, z)$, gives the activity of a topic z, normalized by the sum of activities observed for other keywords, at time-period t.

$$\text{NACT}(t, z) = \frac{\text{ACT}(t, z)}{\sum_{z' \in \mathcal{Z}} \text{ACT}(t, z')}$$

As an illustration of this feature, consider that one monitors a set of topics. With these topics, one describes a particular domain of interest. During one's observation assume that it occurs an event unrelated to this domain of interest. If this event is sufficiently widespread, then it could capture most of available public attention during a short time period. An illustration is as follows, the domain of interest is "quantum computing" thus the monitored topics could be {"qubit","NP-problem","cryptography"}. The event unrelated to "quantum computing" is here the "super-bowl". In such a situation, the decrease of activity for the monitored topics is likely to be uniform. The attention level feature provides, for a domain of interest, the relative importance of each topics that covers the domain of interest. This relative importance withstands such transitory event.

7. **Normalized Number of Users** (NNU). This feature is the counter-part of the normalized activity when ones considers the number of users given by $\text{NU}(t, z)$. It is noted $\text{NNU}(t, z)$ and is defined as follows.

$$\text{NNU}(t, z) = \frac{\text{NU}(t, z)}{\sum_{z' \in \mathcal{Z}} \text{NU}(t, z')}$$

8. **Average Discussion Length** (ADL). This features gives the average length of `discussions` that match the topic z. The definition of this feature noted $\text{ADL}(t, z)$, is as follows.

$$\text{ADL}(t, z) = \frac{\sum_{d \in \mathcal{D}_{t,z}} |d|}{|\mathcal{D}_{t,z}|}$$

The length of disscussion is measured as the number of `content` that it contains.

9. **Activity Change Detection** (ACD). This feature tests, for a topic z, if it exists a change in its mean activity. This test is done with respect to the period that spans from time-period t to time-period t'. There is at most one change point for the

period that spans from t to t'. The output of this feature, noted $\text{ACD}([t;t'],z)$, is a pair $(status, t_c)$. Here *status* belongs to {upward, downward, stable}, the change time noted t_c belongs to $]t;t'[$. To compute this pair one uses CUSUM, as described by [21], in its off-line statistical version. More precisely, Algorithm 1 describes how the *status* value is obtained: once a change point has been identified (by Algorithm 2), one simply compares the mean activity values before and after the change point to establish the proper status value. The change point, if any, is identified by Algorithm 2. This algorithm makes two alternative hypotheses: it exists a change point or it doesn't exist a change point. The likelihood of each hypothesis is computed through a standard maximum likelihood estimation, abbreviated `MLE`, procedure. More precisely, Algorithm 2 compares, for each possible change point $t_c \in]t,t'[$, the likelihood of the two hypothesis that follows, and chooses the most likely one. An illustration of the output of Algorithm 2, when it exists a change, is presented in Figure 2.3. An illustration of the output of Algorithm 2, when it does not exists a change, is presented in Figure 2.4.

- If there is no change point, $\text{ACT}(t_0,z)\ldots\text{ACT}(t,z)$ are consecutive realizations of a unique random normal variable $X_{\eta_{\theta_0}}$ that follows a normal distribution η_{θ_0} of parameters $\theta_0 = (\mu_0, \sigma_0)$;
- If there exists a change at t_c, then the realizations are explained by two distinct random variable $X_{\eta_{\theta_1}}$ and $X_{\eta_{\theta_2}}$ such that $\text{ACT}(t_0,z)\ldots\text{ACT}(t_c,z)$ are realizations of $X_{\eta_{\theta_1}}$, normally distributed according to a normal distribution η_{θ_1} of parameters $\theta_1 = (\mu_1, \sigma_1)$, while $\text{ACT}(t_{c+1},z)\ldots\text{ACT}(t,z)$ are realizations of $X_{\eta_{\theta_2}}$, normally distributed according to a normal distribution η_{θ_2} of parameters $\theta_2 = (\mu_2, \sigma_2)$.

10. **Activity Evolution**. This feature describes the dynamics of $\text{ACT}(t,z)$ using its the first order difference. This feature, noted $\delta_{\text{ACT}}(t,z)$, is defined as follows.

$$\delta_{\text{ACT}}(t,z) = \text{ACT}(t,z) - \text{ACT}(t-1,z)$$

11. **Number of Users Evolution**. This feature describes the dynamics of $\text{NU}(t,z)$ using its the first order difference. This feature, noted $\delta_{\text{NU}}(t,z)$, is defined as follows.

$$\delta_{\text{NU}}(t,z) = \text{NU}(t,z) - \text{NU}(t-1,z)$$

12. **User Engagement Evolution**. This feature describes the dynamics of $\text{UEL}(t, z)$ using its the first order difference. This feature, noted $\delta_{\text{UEL}}(t, z)$, is defined as follows.

$$\delta_{\text{UEL}}(t, z) = \text{UEL}(t, z) - \text{UEL}(t - 1, z)$$

13. **Normalized Activity Evolution**. This feature describes the dynamics of $\text{NACT}(t, z)$ using its the first order difference. This feature, noted $\delta_{\text{NACT}}(t, z)$, is defined as follows.

$$\delta_{\text{NACT}}(t, z) = \text{NACT}(t, z) - \text{NACT}(t - 1, z)$$

14. **Normalized Number of Users Evolution**. This feature describes the dynamics of $\text{NNU}(t, z)$ using its the first order difference. This feature, noted $\delta_{\text{NNU}}(t, z)$, is defined as follows.

$$\delta_{\text{NNU}}(t, z) = \text{NNU}(t, z) - \text{NNU}(t - 1, z)$$

2.3.2 Multiple topics features

The features presented in the previous section do not describe the correlation between the activities of two topics, however such correlations are worthwhile. Consider, for instance, the topics "Santa" and "Christmas", their activities increase similarly during December until the 25^{th} and decrease afterwards. To know that the activities of "Santa" and "Christmas" are increasing can be used to unveil that some other topic, for instance "Barbecue", is not likely to increase. Moreover, the correlations between topics are worthwhile to study events such as the "Super bowl" or the "State of the Union speech". Indeed those events, when they occur, are likely to downsize activities of the topics that aren't related to them. In order to measure such correlations we use the temporal correlation, referred to as *cort*, introduced in [37], and defined below. This correlation captures the dependencies between local trends for two time-series, unlike the Pearson correlation. This correlation is defined for two time-series y_{z_0} and y_{z_1}, that represent the activities observed for a pair of topics. This temporal correlation has values that varies in $[-1, 1]$ where 0 means no correlation between the dynamics of topics, 1 means that topics have the same dynamics, and -1 means that topics have opposite dynamics.

The temporal correlation has a rank parameter r which controls the size of the "memory" available to the process. For instance, setting $r = 7$ means that the temporal correlation is computed with respect to seven consecutives observations. The temporal

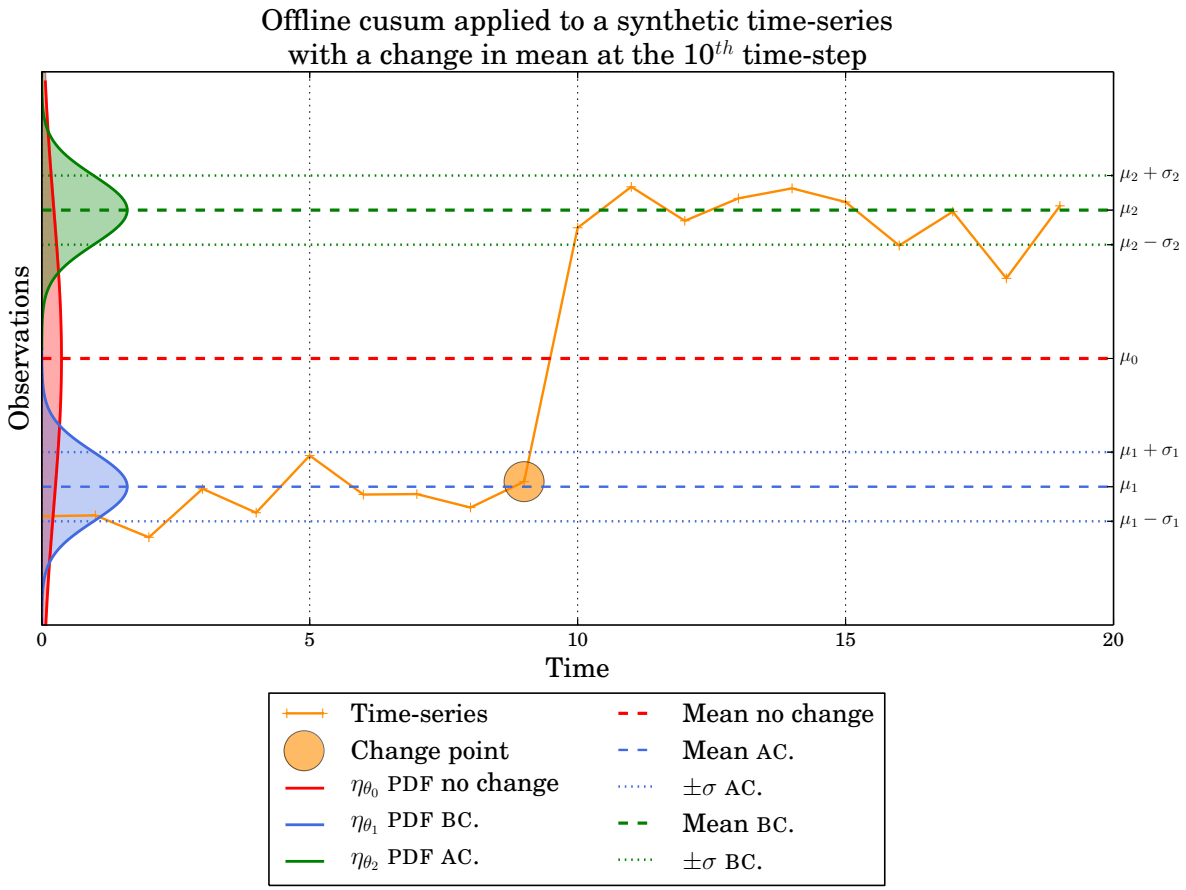

Figure 2.3: Illustration of the Algorithm 2 on a synthetic time-series of 20 observations. The output this algorithm, which is a change time, is illustrated as an orange point. This algorithm compares two hypothesis: it exists a change, or it does not exist a change. In the first hypothesis one assumes that observations are drawn from two normal distributions. Before the change the observations are drawn from $\eta_{\theta_1} = \mathcal{N}(\mu_1, \sigma_1)$ and after the change from $\eta_{\theta_2} = \mathcal{N}(\mu_2, \sigma_2)$. These two distributions are illustrated in blue and green, on the leftmost part of the graph. In the second hypothesis, one assumes that observations are drawn from a single normal distribution $\eta_{\theta_0} = \mathcal{N}(\mu_0, \sigma_0)$, which is illustrated in red. The abbreviations BC. and AC. mean "before change" and "after change", respectively.

correlation is defined with respect to the n-order differences of y_{z_0} and y_{z_1} where n varies from 1 to r. Consequently, the rank parameter has to be set such that n is meaningful for the studied problem. In order to define the temporal correlation one uses the difference of activity for a topic z_i between time-periods t and t', noted $\Delta^t_{t'}(y_{z_i})$, and defined as follows.

$$\Delta^t_{t'}(z_i) = \text{ACT}(t, z_i) - \text{ACT}(t', z_i)$$

In addition, one uses an indexing function, noted $\mathbb{1}_{t,t'}$, that implements the rank parameter by bounding the time-series. This function is defined as follows.

$$\mathbb{1}_{t,t'} = \begin{cases} 1 & \text{if } |t'-t| \leq r \\ 0 & \text{else.} \end{cases}$$

The temporal correlation, for two topics z_0 and z_1, with these two functions, is defined

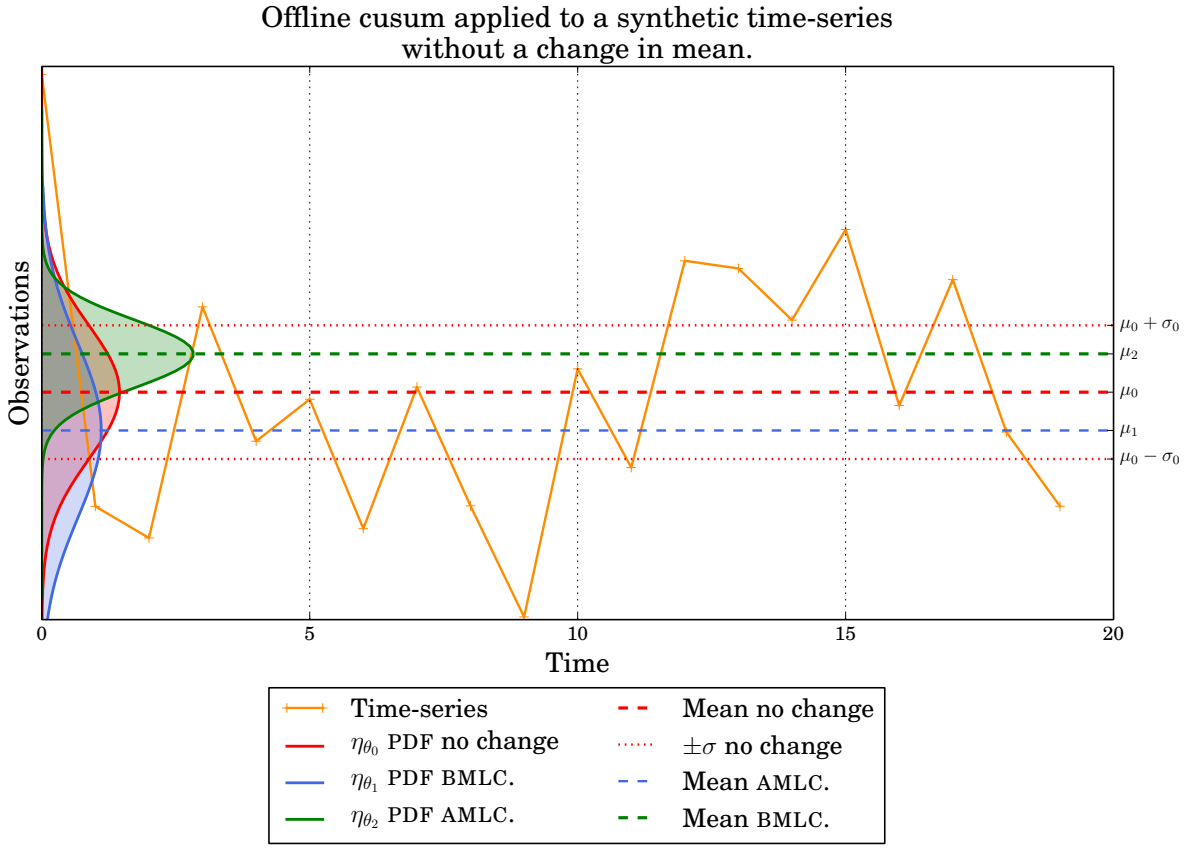

Figure 2.4: Illustration of the Algorithm 2 on a synthetic time-series of 20 observations. As this time-series does not exhibit a change Algorithm 2 does not output a change point. Indeed, it is more likely that all observations are drawn from a single normal distribution $\eta_{\theta_0} = \mathcal{N}(\mu_0, \sigma_0)$, than from two distinct normal distributions. Here the most likely change point is the 10^{th} time-step, hence the parameters of the normal distribution $\eta_{\theta_1} = (\mu_1, \sigma_1)$ are fitted on the 10 first observations, and the parameters of $\eta_{\theta_2} = (\mu_2, \sigma_2)$ are fitted on the 10 lasts observations. As in Figure 2.3, the three normal distributions used in the algorithm are represented on the leftmost part of the graph by the red, green and blue curves. The abbreviations BMLC. and AMLC. mean "before the most likely change point" and "after the most likely change point", respectively.

as follows. The specific details regarding the selection of the parameters r and τ are provided in Section 3.3.2.

$$cort(z_0, z_1) = \frac{\sum_{t,t' \in \mathcal{T}^2} \mathbb{1}_{t,t'} * \Delta^t_{t'}(z_0) * \Delta^t_{t'}(z_1)}{\sqrt{\sum_{t,t' \in \mathcal{T}^2} \mathbb{1}_{t,t'} \Delta^t_{t'}(z_0)^2} \sqrt{\sum_{t,t' \in \mathcal{T}^2} \mathbb{1}_{t,t'} \Delta^t_{t'}(z_1)^2}}$$

This correlation measure is used to define a correlation matrix, this matrix is $n \times n$ if one considers a set of n keywords. This matrix, noted $\mathbf{C}$, is symmetric and has its main diagonal filled with 1. This matrix is as follows: $\mathbf{C}_{i,j} = cort(z_i, z_j)$, hence its i^{th} row, or column, describes the correlation of the activity of topic z_i with the activity of every other topic. Figure 2.5 illustrates such a correlation matrices for three time-series and several values of the rank parameter. One can furthermore weight the matrix $\mathbf{C}$ with the difference of activity observed at a time-period p. The obtained matrix, noted $\mathbf{W}$, can be used to define features that capture activity correlation between keywords with respect to the activity evolution. The definition of this matrix, at time-period p, is as follows.

$$\mathbf{W}^p_{i,j} = \begin{cases} 0 & \text{if } \ i = j \\ \mathbf{C}_{i,j} * \Delta^{p-1}_p(z_j) & \text{else.} \end{cases}$$

The main diagonal of $\mathbf{W}$ is nulled. Otherwise, the features defined with $\mathbf{W}$ would be biased by the correlation of each topic with itself, that is always equal to 1. In addition the matrix $\mathbf{W}$ is pruned in order to keep only the statistically significant values. Except for the main diagonal, the values of $\mathbf{W}$ are distributed according to a normal distribution. The empirical mean and standard deviation of this normal distribution are referred to as: μ_{emp} and σ_{emp}, respectively. In order to obtain a sparse version of $\boldsymbol{W}$ each temporal correlation $\boldsymbol{W}_{i,j}$ that verifies $|\boldsymbol{W}_{i,j} - \mu_{emp}| < 2\sigma_{emp}$ is set to 0, this sparse version is referred to as $\boldsymbol{S}$. Five features are defined with respect to the matrix $\mathbf{W}$, they summarize the distribution of the weighted correlations for each topic. These features correspond to the minimum, maximum and first three moments (mean, standard deviation and skewness) of the distribution of weighted correlation for a topic z_i at time-period p.

1. **Minimum Weighted Correlation** (MWC). The minimum value of weighted correlation for the i^{th} is the minimal value of the i^{th} row of $\mathbf{S}$, at the time-period t. This feature, noted $\text{MWC}(t, z_i)$, is defined as follows.

$$\text{MWC}(t, z_i) = min(\mathbf{S}^t_{i,:})$$

$\text{MWC}(t, z_i)$ is usually negative, there are two situations leading $\text{MWC}(t, z_i)$ to be negative. Firstly, when the activity of a topic z_j has increased, and z_j is inversely correlated to z_i (*ie.* $\mathbf{C}_{i,j} < 0$). Secondly, when, the activity of topic z_j has decreased,

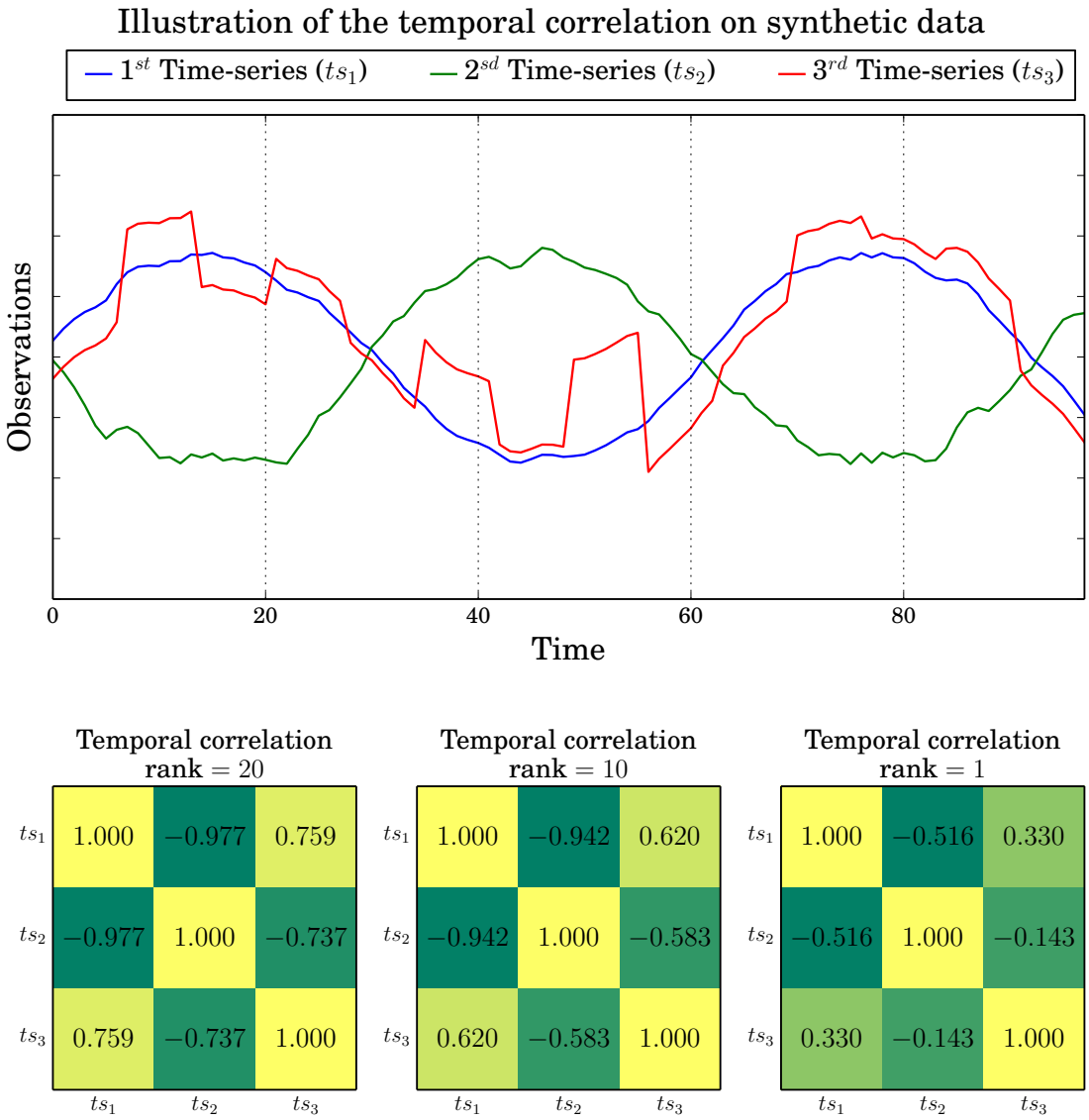

Figure 2.5: Illustration of the temporal correlation computed for three synthetic time-series named ts_1, ts_2, and ts_3. This Figure presents three temporal correlation matrices, for three different values of the rank parameter: 20, 10 and 1. The second time-series is roughly the opposite of the first one, hence the temporal correlation are negative and close to -1 when the rank parameter is high enough. The third time-series is made from the first one and a normal noise that change once every 7 time-steps. Therefore, when the rank parameter is greater than 7 the temporal correlation has values higher than 0.5.

and z_j is correlated to z_i (*ie.* $\mathbf{C}_{i,j} > 0$).

2. **Maximum Weighted Correlation** (XWC). The maximum value of weighted correlation for the i^{th} is the maximal value of the i^{th} row of $\mathbf{S}$, at the time-period t, this feature, noted $\text{XWC}(t, z_i)$, is defined as follows.

$$\text{XWC}(t, z_i) = max(\mathbf{S}^t_{i,:})$$

$\text{XWC}(t, z_i)$ is usually positive, there are two situations leading $\text{XWC}(t, z_i)$ to be positive. Firstly, when the activity of a topic z_j has decreased, and z_j is inversely correlated to z_i (*ie.* $\mathbf{C}_{i,j} < 0$). Secondly, when the activity of topic z_j has increased, and z_j is correlated to z_i (*ie.* $\mathbf{C}_{i,j} > 0$).

3. **Average Weighted Correlation** (AWC). This feature, noted $\text{AWC}(t, z_i)$, is defined as follows when one considers a set of n topics.

$$\text{AWC}(t, z_i) = \frac{1}{n} \sum_{j=0}^{n} (\mathbf{S}^t_{i,j})$$

4. **Standard Deviation of the Weighted Correlation** (STDWC). This feature, noted $\text{STDWC}(t, z_i)$, is defined for the topic z_i, when ones considers a topic set of size n, as follows

$$\text{STDWC}(t, z_i) = \sqrt{\frac{1}{n} \sum_{j=1}^{n} (\mathbf{S}^t_{i,:} - \text{AWC}(t, z_i))^2}$$

5. **Skewness of the Weighted Correlation** (SKEWC). This feature, noted $\text{SKEWC}(t, z_i)$, is a measure of the asymmetry of the weighted correlation of the topic z_i. This feature is defined as follows when one consider a set of n topics.

$$\text{SKEWC}(t, z_i) = \frac{\frac{1}{n} \sum_{j=1}^{n} (\mathbf{S}^t_{i,j} - \text{AWC}(t, z_i))^3}{\left(\frac{1}{n} \sum_{j=1}^{n} (\mathbf{S}^t_{i,j} - \text{AWC}(t, z_i))^2 \right)^{3/2}}$$

The features proposed in Section 2.3.1 and Section 2.3.2 are defined with respect to an unspecified topic model. The choice of a topic model depends on the envisaged application as well as the applicative restrictions. As presented in Section 2.1.5, in our industrial setting the topic model is a simple lookup in a product-data-base. This look-up takes has input all the tokens of every user-generated-content. Indeed the envisaged

application is to forecast the customers activity about a set of products, brands and products lines. However, as presented in Section 2.1.5, numerous more sophisticated topic model are usable. In the next section the feature presented above are used to tackle the different activity prediction problems defined in Section 2.2. The forecast capacities are studied and several experiments are realized to determine which are the factors that improve or decrease the results of the activity prediction problems.

Conclusion

This chapter introduces a generic framework that is used to describe various social-media. With the generic framework one can define *social-media-mining* problems and features to tackle these problems. In this chapter three activity prediction problems are proposed. In addition features based on the generic framework are defined to tackle these three activity prediction problems.

The generic framework proposed in Section 2.1 can describe various social-media. For instance, Twitter as presented in Section 2.1.2, Facebook as presented in Section 2.1.3, and bulletin board system as presented in Section 2.1.4. The generic framework uses topic-models to group together user-generated-contents that are related to each others. The generic framework does not enforce a particular topic model. As presented in Section 2.1.5, there are numerous of such topic models, and the one used in this study amounts to a simple look-up in a product-data-base for each token of every user-generated-content. This simple method is chosen with respect to industrial objectives. These objectives are to be able to forecast the interest of customers for the items of a product-data-base. The generic framework is used to define features that are used to tackle the activity prediction problems. Therefore, these features can be adapted to any social media that fits in the proposed generic framework. The generic framework is minimalistic and describe most of the popular social-media. Indeed, this representation relies on public retrievable data. Indeed, private data are sold by the companies operating the social media at unacceptable prices. The user-network associated to a social media, as presented in Section 1.2.1, is not described in the generic framework. Therefore, features such as the centrality of a user in the user-network, or similar popularity measures, can not be defined with the generic framework.

The features proposed in Section 2.3 to tackle the activity prediction problems are

divided in two groups. Firstly, the features that describe one single topic. Secondly, the features that are based on the temporal correlation between topics weighted with the variation of activities for each topic. These features are not comprehensive, and various other features could have been devised. For instance, features based on the content of the messages, or features based on users. That said, specific challenges arise with such features. The features based textual content rely on natural language processing. Thus in order to use such features one has deal with social-media-related issues such as: lack of context; presence of misspelled words or domain specific acronyms; detection of sarcasm; processing of several distinct languages. Despite these issues some features such as the presence of a URL or the arousal (*ie.* excitement) level have been studied for instance per Morchid *et al.* [116].

The next chapter presents the data collection, and several experiments. These experiments aims at evaluate if the proposed features allows to use state-of-the-art machine learning methods in order to solve the activity prediction problems. In addition, the next chapter presents the collection and preparation of data from Twitter and a bulletin-board-system.

Data: $\mathcal{Z}$
Result: a status $\{upward, downward, stable\}$ per topic $z \in \mathcal{Z}$

forall the $z \in \mathcal{Z}$ **do**
  observed_activity $\leftarrow \{\text{ACT}(t_0, z), \ldots, \text{ACT}(t, z)\}$;
  change_time $\leftarrow$ `cusum_change_time`(observed_activity);
  if *change_time* $\neq 0$ **then**
    Activity_before_change $\leftarrow \{\text{ACT}(t_0, z), \ldots, \text{ACT}(change\_time, z)\}$;
    Activity_after_change $\leftarrow \{\text{ACT}(change\_time + 1, z), \ldots, \text{ACT}(t, z)\}$;
    if *mean(Activity_before_change)* $<$ *mean(Activity_after_change)* **then**
      `label` z as *downward*;
    else
      `label` z as *upward*
    end
  end
  if *z has no status* **then**
    `label` z as *stable*
  end
end

Algorithm 1: Activity Change Detection Feature: This algorithm classify each topic in $\mathcal{Z}$ with respect to it past activity level. Each topic is classify in exactly one category: stable, upward or downward. This classification is based on Algorithm 2 that extract the most likely time-period of change in the mean activity. If Algorithm 2 does not output a change point for a topic, then this topic is considered as stable. On the contrary if a change point is found, the topic is label according to the change in its activity level.

Data: $\mathcal{A} = \{\text{ACT}(0, z), \ldots, \text{ACT}(n, z)\}$ the activity levels of topic z observed between the time-period p_0 and the time-period p_n.

Result: change_point a time-period index between 0 and n of the most-likely change in activity level.

$\theta_0 \leftarrow \texttt{MLE}(\mathcal{A})$;
likelihood_no_change $\leftarrow \prod_{\tau=0}^{n} p_{\theta_0}(\text{ACT}(\tau, z))$;
likelihood_change $\leftarrow 0$;
change_point $\leftarrow 0$;
forall the $t \in [1, n-1]$ **do**
    $\theta_1 \leftarrow \texttt{MLE}(\text{ACT}(t, z), \ldots, \text{ACT}(n, z))$;
    likelihood_change_at_t $\leftarrow \prod_{\tau=0}^{t} p_{\theta_0}(\text{ACT}(\tau, z)) * \prod_{\tau=t+1}^{n} p_{\theta_1}(\text{ACT}(\tau, z))$;
    if *likelihood_change_at_t* $>$ *likelihood_change* **then**
        likelihood_change $\leftarrow$ Likelihood_change_at_t;
        change_point $\leftarrow$ t';
    end
end
if *likelihood_no_change* $\leq$ *likelihood_change* **then**
    return *change_point*
else
    return *0*
end

Algorithm 2: Offline Cusum Algorithm: This algorithm presents the unsupervised change detection referred to as `cusum_change_time`. This change detection is performed with the cusum offline change in mean detection. This algorithm is used in Algorithm 1 which compute a feature based on the change detection. The standard maximum likelihood estimation is abbreviated `MLE`.

Chapter 3

Activity prediction

Contents

Introduction

This chapter presents several experiments related to the prediction problems presented in Section 2.2. This chapter also describe how the data used in these experiments is collected and prepared. The data is collected in two social-media. The first social-media is Twitter, it is most widely used micro-blogging system. The second social-media is a bulletin-board-systems operated by a company named Purch. This company operates numerous web-sites and bulletin-board-systems. Most of these bulletin-board-systems are dedicated to computer hardware, video games, and mobile telephony. The most known bulletin-board-system operated by Purch is Tom's hardware [1]. Purch employs editors to cover news-stories and benchmark products. Purch incomes depends on its ability to sell advertising space. This study is funded and supported by Pruch. Hence, privileged data, such as the page-views quantities, were available for the bulletin-board-system.

The first section of this chapter presents the collection of the data for Twitter and the bulletin-board-system. The data collection is an automated daily routine. The collection of data spans over 51 weeks. The data that is collected matches a list of terms provided by Purch. This list of terms is based on the user-generated-contents observed on the bulletin-board-systems operated by Purch. The data collection on Twitter uses the public API of Twitter and therefore is subject to some restrictions. The data collection is monitored to correct technical failures. Collected data are filtered to remove commercial contents (*ie.* spam). Indeed the social-media are targets for the diffusion of commercial contents at large scale. A text-based classifier is trained to filter out the user-generated-contents that prove to be commercial content. Indeed, these commercial contents might induce noise in the observations.

The second section of this chapter presents how collected data were used to define training sets for the three activity prediction problems defined in Section 2.2. More precisely, this section defines a cross-validation method that takes into account the time of observations. Indeed, the problems that are studied are time dependent and classical cross-validation methods might not evaluate correctly the predictive capacities. This section also presents an unsupervised method to extract buzz candidates that is based on the activity levels. The buzz candidates are topics which are likely to have an important increase of their activity. To extract buzz candidates is necessary to balance the training-sets of the buzz classification problem. Indeed very few topics are subject to

an important activity increase, therefore most instances are negative examples for the buzz classification problem. Some of the terms provided by Purch are ambiguous. The influence of ambiguity on the activity prediction is an important matter. In order to study the effects of ambiguity, three lists of terms are defined, each one with a different ambiguity level. This section presents a method to evaluate the ambiguity level of a term that is based on wikipedia.

Each of the three remaining section of this chapter is dedicated to the study of one activity prediction problem. The third and fourth sections present the buzz classification and the magnitude prediction, respectively. These problems are studied within two different experimental settings. The first experimental setting allows to compare the results on Twitter and the bulletin-board-system, on a long time-scale, and with three different languages. On the contrary the second experimental setting is dedicated to Twitter where the activity proves to be more difficult to predict. This second experimental aims to increase the reproducibility as much as possible. Hence, this experimental setting includes user-generated-contents collected with respect to a list unambiguous English terms. In addition the observation are restricted to ten consecutive weeks. Hence the risk of change in the public API of Twitter is lowered. The last section of this chapter studies the rank prediction problem solely for the second experimental setting.

3.1 Data collection

In order to evaluate the activity prediction, we collected data from Twitter on a daily basis. These data are completed with data extracted from a bulletin-board-system. The data collection lasted for 51 weeks from October 2011 to October 2012. On Twitter, the data collection is done by the mean of the REST API. The user-generated-contents were retrieved according to a list of terms. This list is provided by *Purch* the company that supported and funded this project. *Purch* built this list of terms from the user-generated-content observed in their bulletin-board-system. There are 6671 terms in this list. Some of these terms contains several words, for instance: “hewlett packard” or “home server”. There are several types of terms in this list: brand names (*eg.* Microsoft); technology names (*eg.* lcd monitor); product-line names (*eg.* dell inspiron, ipod nano). All these terms are related to high technology matters. Indeed the bulletin-board-system from which the terms were extracted are dedicated to high technology discussions [1]. The terms comes from three different languages: English, French and German. During the

data collection, the user-generated-contents that contain at least one item from the list of terms were retrieved. As discussed in Section 1.2.4, the Twitter REST API has usage limitations. These limitations cannot be bypassed easily. Therefore some user-generated-contents can be missing from the dataset. In total 588 millions user-generated-contents were collected. These user-generated-contents were produced per 48 millions distinct users. A user-generated-content might be observed multiple times if it contains several items of the list of terms. The user-generated-contents that were observed more than once are discarded. It is straightforward to discard a user-generated-content that is observed multiple times, indeed each user-generated-content is provided with a unique numerical identifier.

3.1.1 Detection and removal of commercial contents

As pointed out by Kurt *et al.* [149], Twitter is not spared from commercial contents, or "spam". As the creation of commercial content is often automated and large scaled is it necessary to remove the commercial contents beforehand. Benevenuto *et al.* [22] studied the detection of commercial contents in Twitter. In their study the objective is to detect the users that produce commercial contents. Hence, key features are the number of followers, the age of the account, and features based on the content of tweets. Another approach is to classify each user-generated-content based on its textual content. Indeed, a key finding from previous studies is that URLS are much more frequent in commercial content. Similar finding can be expected for some specific tokens such as: "*$; discount; deal; shipping*". There are two frequent patterns of commercial content in the collected data. The first one aims at spread a promotion for a particular product, as for instance: "*[$] black friday cheap price viewsonic va1906a led 19 inches t widescreen led monitor* ⟨URL⟩". On the contrary, in the second pattern, the commercial content aims at increase the visibility of the content-producer. In order to manage to do that, the commercial contents are built as "contests" in which users are invited to forward a message so they have a chance to win a prize *eg.* "*Follow @WinObs, and RT this for a chance to win a Xbox 360 Wireless Wheel* ⟨`url`⟩ *#giveaway #contest*".

The detection of commercial content is treated as a binary classification problem where each user-generated-content is labeled either as commercial content or standard content. Two binary classifiers are trained. Each classifier is trained with a single pat-

tern of commercial content. The outputs of these two binary classifiers are merged. A user-generated-content that is labeled as commercial content by at least one of the two classifiers, is discarded. In order to propose a scalable solution, the two classifiers are based solely on the textual content of user-generated-content. The textual contents are described as "bag-of-words" and each word defines a feature using a one-hot encoding. The user-generated-content are preprocessed as follows: English stop-words and punctuation are removed, the URLS and the user-mentions (*eg.* @user) are replaced with dedicated tokens: ⟨`url`⟩ and ⟨`mention`⟩, respectively. This avoids to consider each URL or user-mention as an extremely rare word. The one-hot encoding is applied to the token of the user-generated-content. This implies, for a support vector classifier, that a user-generated-content which has none of the tokens observed in the training-set is labeled as standard content. Indeed such an user-generated-content is represented as a vector of zeros. This behavior reduces the risk of wrongly label a standard content as commercial. Around 2000 user-generated-contents were hand-labeled in order to train the two binary classifiers. There are 4181 different tokens for the binary classifier that is trained with the promotions, and 4775 different tokens for the binary classifier that is trained with the contests. The training sets are balanced with as many positive instances than negative instances. Performances are evaluated with a 10-folds cross-validation. The classifiers are support vector machines with a linear kernel. For each fold the train part is used to fit the hyper-parameters of the support vector machine with a grid search, then the training itself is done with the best hyper-parameters. The average results of the cross-validation are reported in Table 3.1. We applied these classifiers to the whole dataset and 6.1 millions tweets were removed from the dataset, an acceptable result.

3.1.2 Collection failure monitoring

Lastly, the amount of user-generated-content collected per time-period is monitored. In order to be more resilient to technical failures, the abnormal time-periods are interpolated with their previous and next time-periods. The detection of abnormal time-periods is based on the number of terms for which none user-generated-content were captured. More precisely, the outlier detection is based on the standard score. When an abnormal time-period is detected the activity is interpolated independently for each term. There were only two abnormal time-periods over a 51 weeks collection period.

	Promotion		Contest	
	positive	negative	positive	negative
Precision	0.968	0.970	0.948	0.946
Recall	0.993	0.982	0.950	0.948
F1-score	0.980	0.975	0.949	0.946

Table 3.1: Results of the cross-validation for the detection of commercial contents. The cross-validation is 10 folds. The reported results are average over all the folds. The left column of this table presents results obtained for the training-set of user-generated-contents with a promotion on a particular product. The right column of this table presents results obtained for the training-set of user-generated-contents with an invitation to a contest.

3.1.3 Overview of the collected data

The bulletin-board-system and Twitter are of different nature. Twitter is a fast-paced social-media in which the messages are organized with respect to their author as presented in Figure 2.1.2. On the contrary the bulletin-board-system are slow-paced. Moreover, in a bulletin-board-system, the messages are archived in a hierarchical structure described in Figure 2.1.4. The collection of data on these two social-media results in datasets of different nature. Table 3.2 reports a summary of the collected data for both social-media. The collected data is used to make different data-sets, some of which were published on the UCI machine learning repository [2].

	Nuber of users			Number of `discussions`		
	FR	EN	DE	FR	EN	DE
BBS	$72{\cdot}10^3$	0	$8{\cdot}10^3$	$50{\cdot}10^4$	0	$1{\cdot}10^4$
Twitter	$24{\cdot}10^6$	$30{\cdot}10^6$	$10{\cdot}10^6$	$232{\cdot}10^6$	$287{\cdot}10^6$	$46{\cdot}10^6$

Table 3.2: Summary of the collected data on Twitter and the bulletin-board-system detailed per language. The abbreviations FR, EN, and DE stand for French, English, and German, respectively. The abbreviation BBS stands for bulletin-board-system. The left column of this table report the number of discussions as presented in Definition 8.

3.2 Creation of the training-sets

The training-sets for the magnitude prediction and the ranking prediction are generated with respect to a history size and forecast horizon. These two parameters, named α and δ respectively, are presented in equations 2.1 and 2.5. The training-set for the buzz classification depends on two additional parameters. These parameters are named σ and φ. The parameter σ controls the labeling function, as presented in equations 2.3 and 2.4. The smoothing parameter φ controls the extraction of buzz candidates that is presented in Section 3.2.2. The partition of the training by a train-test split is now presented without specifying values for this parameters. A train-test is used to split in two parts the available data in order to evaluate the prediction. The first part of the data is considered as known or observed. On the contrary the remaining of the data is considered unknown or unobserved. The observed data is used to train a model. Once a model trained, the unobserved data is used to test the model trained previously. The classical methods to do a train-test split, as for instance leave-one-out, or n-folds, do not take into account the temporal order of the instance. Hence, the future with respect to a train instance is known when it belongs to the train part. Such a situation could bias the evaluation process, and we propose a train-test split that takes into account the time.

3.2.1 Train-test split

In order to get a train-test split that takes time into account, the split is based on the time-periods that are defined in Section 2.1.1. A time-period $t_i \in \mathcal{T}$ is considered along with the $m+1$ time-periods that surround it. More precisely the considered time-periods are as follows: $\{t_{i-m}, \ \dots \ , t_i, t_{i+1}\}$. The time-periods $\{t_{i-m}, \ \dots \ , t_{i-1}\}$ are the past with respect to t_i. On the contrary the time-period t_{i+1} is the future with respect to t_i. The time-periods that belong to the past are considered as observed. On the contrary, the time-period that belongs to the future is considered as unobserved. Every pair of consecutive time-periods in the past contributes to the train-set. The pair of time-periods t_i and t_{i+1} is the test-set. The train and test sets are tuples of form $\langle \boldsymbol{X}, Y \rangle$. In this tuple $\boldsymbol{X}$ is the matrix of features and Y is a vector of labels. The features in $\boldsymbol{X}$ are presented in Section 2.3, and $\boldsymbol{X}_{i,j}$ is the value of the jth feature for ith topic during the considered time-period. The labels are computed according to the problems of interest that are described in Section 2.2. For instance, one that considers the Buzz classification uses one of the three labeling functions defined in equations 2.2, 2.3 or 2.4. Figure 3.1 illustrate the train-test split with four train sets: $\langle \boldsymbol{X}_{train\ 0}, Y_{train\ 0} \rangle \ \dots \ \langle \boldsymbol{X}_{train\ 3}, Y_{train\ 3} \rangle$. These

train sets are such that $Y_{train\ j}$ is made with time-period t_j and $\boldsymbol{X}_{train\ j}$ with time-period t_{j-1}. The Algorithm 3 presents a pseudo code of this train-test split.

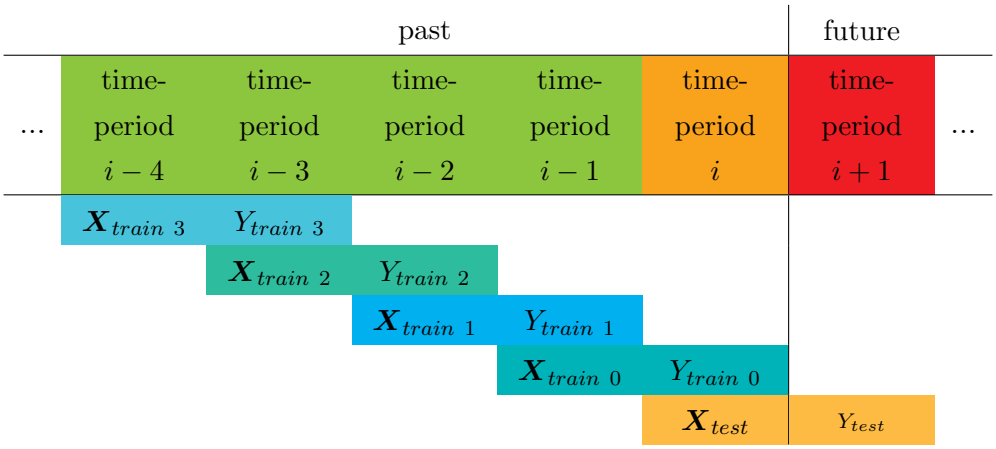

Figure 3.1: Illustration of the train-test split that take time in account. The data from the time-period p_{i+1} is considered as unknown. The data from time-periods before p_i included are considered are observed and used to compute alternatively features or labels.

The above explanations describe a simplified view of the actual train-test split. Indeed one might want to consider more or less historical data to build the feature matrix $\boldsymbol{X}$. In the same way, one might want to vary the forecast horizon. Hence, in the actual train-test split, the time periods can be aggregated in order to match the applicative needs.

3.2.2 Extraction of buzz candidates

Most items of the term list are not prone to have an activity burst. Indeed most items are not widely popular, hence their activity level is feeble and almost constant over time. To consider all these latent terms produces an highly unbalanced training-set for the buzz classification problem. An ill-balanced training-set with mostly negative instances result in a biased, unusable, buzz classifier. It is therefore necessary to filter out these latent terms in order to focus on those terms that are in the early stages of a dramatic activity increase. In order to detect the early stages of the activity increase, a change detection method is used. This change detection is applied to activity levels of the time-period that predate the prediction. When an activity-change is detected, then the mean activity before the change and after the change are compared. If this activity level is stable or decreasing, then it is highly unlikely that the keyword has a dramatic activity increase

Data: a sequence of time-periods $\mathcal{T} = [t_0, \ldots, t_{max}]$ and an integer m
Result: a sequence of pairs $[(train; test)_m, \ldots, (train; test)_{max-1}]$

forall the $t_i \in [t_m, \ldots, t_{max-1}]$ **do**
  $k \leftarrow (m+1) \mod 2$;
  observations $\leftarrow [t_{i-m}, \ldots, t_i]$;
  $\boldsymbol{X}_{train} \leftarrow \texttt{features}(\text{observed}_0)$;
  $Y_{train} \leftarrow \texttt{labels}(\text{observed}_1)$;

  forall the $t_j \in observations_{1:i-k}$ **do**
    $\boldsymbol{X}_{train} \leftarrow concatenate(\boldsymbol{X}_{train}, \texttt{features}(t_j))$;
    $Y_{train} \leftarrow concatenate(Y_{train}, \texttt{labels}(t_{j+1}))$;
  end

  $train \leftarrow \langle \boldsymbol{X}_{train}, Y_{train} \rangle$;
  $test \leftarrow \langle \texttt{features}(t_i), \texttt{labels}(t_{i+1}) \rangle$;
  Yield $(train, test)_i$
end

Algorithm 3: This algorithm splits the collected data into in a sequence of train and test sets. The functions $\texttt{features}(t_i)$ and $\texttt{labels}(t_i)$ provide, for the time-period t_i, the labels that are presented in Section 2.2 and the features that are presented in Section 2.3. The function $concatenate(x, y)$ stacks the matrices x and y of identical shape (n, m) into a single matrix of shape $(2n, m)$.

during the time-period of evaluation. As a consequence, the term is discarded for this time-period. On the contrary, if the activity level shows an upward tendency the term is added to the set of buzz candidates. This set contains terms that might have a dramatic activity increase during the time-period of evaluation. The change detection method is based on CuSum proposed by Page [123]. This unsupervised change detection method is comprehensively described in the book of Basseville and Nikiforov [20]. This method has been shown to be effective in different contexts for instance with data from stock markets [166]. This method is also used to defined the ACD feature presented in Section 2.3.1. Figures 2.3 presents the output of CuSum on synthetic data which a significant change. On the contrary Figure 2.4 presents the output of CuSum on synthetic data without a significant change. The selection of buzz candidates is now presented in details.

For each term z and time-period t we consider the activity level of z during a time-interval $[s, s']$ that varies with respect to t. This activity level is $Y(z, [s; s'])$, as presented in Section 2.2. The activity level is segmented with CuSum. If none segmentation point is found, the activity level is deemed constant and the term z is filtered out for the time-period p. When a segmentation point is identified at time $s + \beta$ the values of Y for the two segments are compared:

- If $\texttt{average}(Y(z,]s + \beta; s'])) < \varphi \times \texttt{average}(Y(z, [s; s + \beta]))$, then we consider that the term z does not display an upward tendency. Thus, the term z is filtered out for time-period t_i. The smoothing parameter φ allows to configure the strength of the upward tendency needed for a term to be considered as a buzz candidate.
- Otherwise, the term z is added to the set of buzz candidates for the time-period t_i.

Despite this filtering, the training-set for the Buzz classification problem is still biased toward negative examples. Detailed information about the balance of the classes in the training-set are provided in Section 3.3. Once buzz candidates have been extracted, they can easily be annotated with one of the labeling functions defined in Equations 2.2, 2.3 and 2.4. Figure 3.2 presents three terms with two buzz candidates and one terms that is filtered out.

3.2.3 Ambiguity consistent data-sets

In order to evaluate the effect of the ambiguity of terms, three lists of terms with comparable ambiguity are created. These three lists are based solely on the English terms. Each list of term is used to produce a data-set. To evaluate the influence of the ambiguity is an important question. Indeed, to disambiguate terms used in user-generated-contents is a difficult task. The key problems for the disambiguation applied to user-generated-contents are the lack of context and the usage of domain specific expressions or spellings. DBpedia Spotlight is a reliable *ad-hoc* disambiguation method based on a semantic version of wikipedia. We tested DBpedia Spotlight on 250 tweets that were uniformly sampled from the Twitter-data in English. In total 331 ambiguous terms are labeled manually. Out of these 331 ambiguous terms 209 (63.1%) are correctly disambiguated by DBpedia Spotlight. The most ambiguous terms constitute the majority of disambiguation errors. It confirms that the disambiguation in Twitter has to be addressed by specific methods.

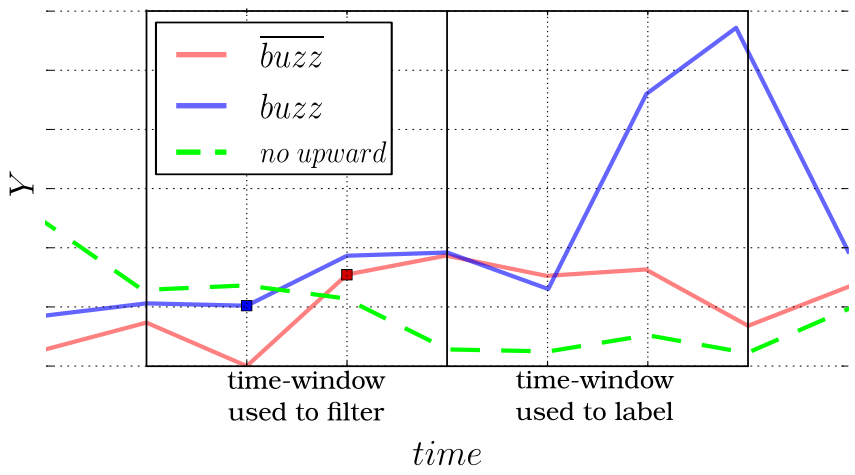

Figure 3.2: This illustration shows three terms that are submitted to the buzz candidate filtering. The dashed line illustrates a term without any upward tendency that is filter out. The two other lines illustrate buzz candidates. Among the buzz candidates, the one represented with a blue line turns out to be a positive instance. On the contrary the red line illustrates a buzz candidate that turns out to be a negative instance. The segmentation point is presented as a square.

In order to evaluate the effect of the ambiguity of terms, three lists of terms with different ambiguity levels are created. Each list of term is used to produce a data-set. A simple representation of the ambiguity is used. The ambiguity of a term is equal to the number of meanings it has in wikipedia. The wikipedia disambiguation pages list all the possible meanings of a term. The number of entries in the wikipedia disambiguation pages are counted to measure the number of meanings that has a term. The three lists of terms are created as follows. Firstly, the LOW list contains only unambiguous terms which have one single meaning. Secondly the MED list that contains terms with 3 to 19 meanings. Finally the HIGH that contains terms with 20 to 99 meanings. Figure 3.3 presents the distribution of ambiguity level in the MED list and the HIGH list. In addition this figure presents the activity levels for the three lists of terms.

The distribution of the ambiguity level in the list of terms MED is very skew, with almost half of its terms have three possible meanings. On the contrary the list of terms HIGH is more uniformly distributed. The aggregate activity per week and term varies for the three lists of terms. The LOW list contains fewer highly active terms than MED and HIGHwith. Approximately half (50,6%) of the terms in the list LOW have less than 500 user-generated-contents per week. On the contrary, this quantity drops to 12% for the

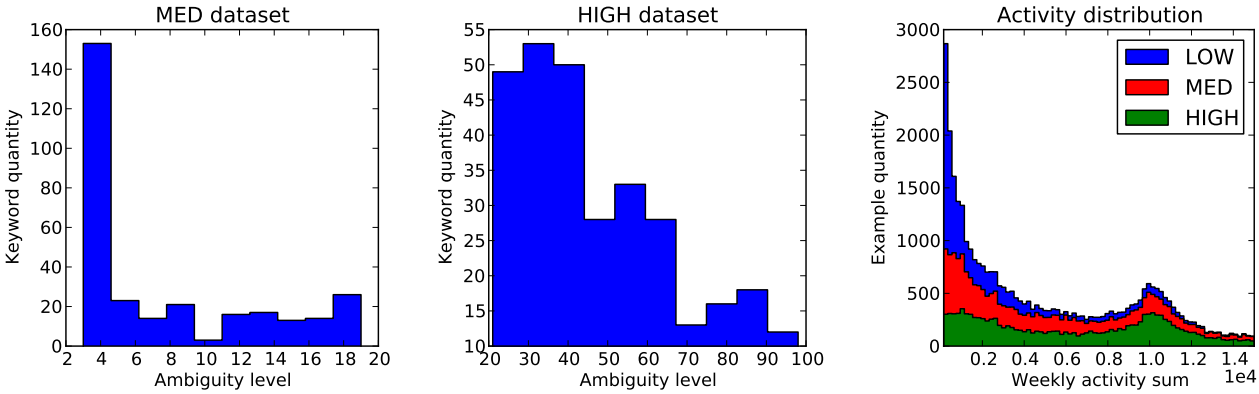

Figure 3.3: This figure presents the distributions of the ambiguity levels based on wikipedia. The leftmost part of this figure presents an histogram of the ambiguity level for the terms in the MED list. The central part of this figure presents an histogram of the ambiguity level for the terms in the HIGH list. The rightmost part of this figure presents three stacked histograms of the activity per list of terms. The activity is measure with the ACT feature and aggregated over one week.

list MED, and 5% for the list HIGH.

3.3 Buzz classification

The section is devoted to the evaluation the buzz classification problem. This problem is tackled with state-of-the-art machine learning methods and the features presented in Section 2.3. Several parameters are required in order to build the training-set. The choice of these parameters is discussed below and various applicative scenario are tested.

3.3.1 First experimental setting

The buzz classification is made with respect to a measure of the activity per term and time-period. In this first series of experiment, two of such activity measures are considered. For twitter, the measure of activity is the number of active discussions. This measure is abbreviated NAD and defined in Section 2.3.1. For the bulletin-board-system, the measure of activity is the number of page-views per term and time-period. This measure is abbreviated ND. This measure is rarely available and it is hard to approximate it. The number of page-views is equal to the number of visits that a web-resource receives per time-period. For a term z, the number of page-views is aggregated for all the

web-resources that contains a `discussion` which match the term z. In this experimental setting, the list of 6671 terms presented in Section 3.1 is used.

The proposed approach makes use of several parameters which have to be fixed. First of all, the size of a time-period is fixed to one day, indeed we don't plan to do work with a time-resolution finer than a day. For twitter the matrices of features $\boldsymbol{X}$ are based on seven time-periods, and the label vectors $\boldsymbol{Y}$ are based on two time-periods. In this situation one aims at predict if a buzz will occur in the following couple days based on the past week. For the bulletin-board-system the matrices of features are base on sixty time-periods, and the label vectors are based on fourteen time-periods. This choice is motivated by the specificities the two social-media. Indeed, the bulletin-board-system have fewer users than twitter and the number of user-generated-contents created per day is noticeably lower than in Twitter. The selection of the buzz candidates is based on the time-periods used as train. Hence, in twitter the time-windows submitted to CuSum last seven days. In the bulletin-board-system the time-windows last sixty days. The selection of the buzz candidates has one parameter. This parameter a smoothing parameter named φ, it controls the strength of the upward tendency that is required for a term to be considered as a buzz candidate. The smoothing parameter φ is set to 1.25. Hence, a term is a buzz candidate when its average-activity increases of a quarter after the segmentation point detected by CuSum. The second parameter to set is the threshold named σ that controls the labeling functions defined in equations 2.3 and 2.4. In order to cover various applicative scenarios, several values of σ are tested within the range $[500, 4500]$. This range of values is also used by Suh *et al.* [145]. In total 140,707 buzz candidates were extract from the Twitter-data, and 7,905 form the bulletin-board-system-data. The classes are strongly unbalanced towards negative example for the two social-media, regardless of value of the threshold σ. In the worst case scenario, the ratio negative example out of positive example is higher than 100. Figure 3.4 presents the balance of the classes for the two social-media and several values of the parameter σ. It is noteworthy that the relative growth approach, defined in equation 2.4, generates fewer positive examples. The features that are define in Section 2.3.2 are not used in this experimental setting, indeed, the features were defined after the completion of these experiments.

The performance of three state-of-the-art approaches are compared. These approaches are: random forests, support vector machines (Abbr. SVM) and k-nearest-neighbors (Abbr. k-NN). The meta-parameters of these approaches are estimated by a grid search

procedure. A grid search procedure is a sequential test of all the possible combinations for the chosen values of the parameters. For k-NN, a single parameter is tuned. This parameter is the number of neighbors to consider. The number of neighbors to consider varies in $\{1, 3, 5, 7, 9, 13, 17, 19, 100\}$, and the Euclidean distance is used for all the tests. For random forests, the number of trees varies from 1 to 100 with a step of 3. The other meta-parameters are fixed as proposed by Breiman [32]. Hence, parameters such: the maximum tree depth, or the number of feature to sample, are not tuned. For SVM, a RBF kernel is used. The class weight is set with the value of the ratio of positive examples out of negative ones. The parameter C varies in $\{10^{-2}, 10^{-1}, 1, 10, 10^2, 10^3, 10^4, 15000, 20000, 50000\}$ and the parameter γ varies in $\{10^{-3}, 10^{-2}, 10^{-1}, 1, 2, 5\}$. The readers interested in a comprehensive description these parameters are referred to the book by Murphy or Bishop [117, 24]. To select these meta-parameters on the whole collected data has a great computational cost. Hence, one fifth of the data is uniformly sampled to select the meta-parameters. This sample has approximately 28,000 examples. Table 3.3 presents three boradly used measures of the performances a binary classifier. In addition, this

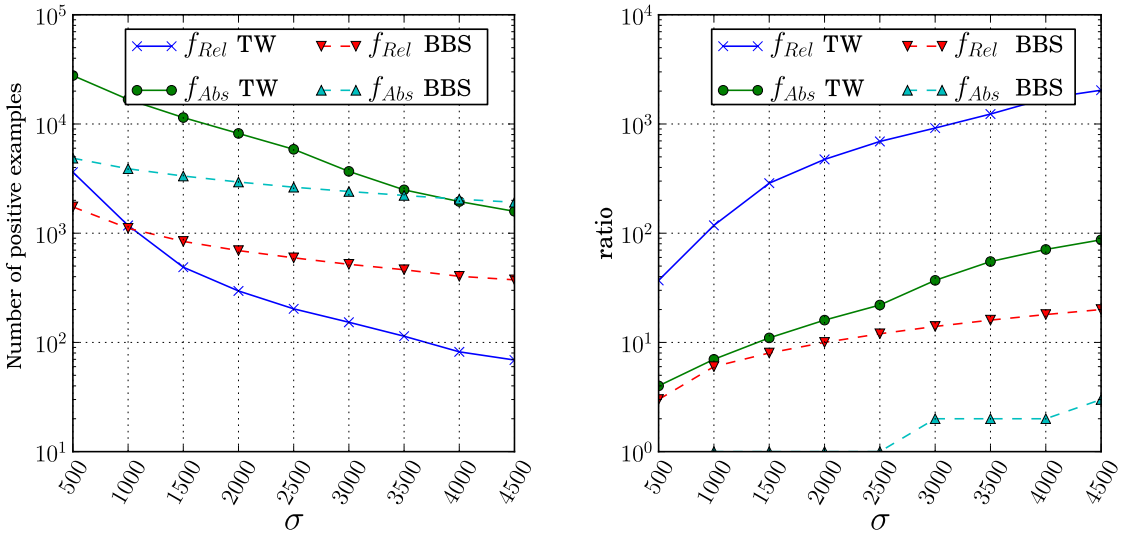

Figure 3.4: This figure illustrates the balance of the classes for several values of the parameter σ. The left side of this figure shows the number of positive example. The right side of this figure shows the ratio of positive negative example out of positive ones. This two plots have a logarithmic scale. The labeling function defined in Equation 2.3 is abbreviated f_{abs}. The labeling function defined in Equation 2.4 is abbreviated f_{rel}. Twitter and the bulletin-board-system are abbreviated TW and BBS, respectively.

table presents the training runtime, expressed as a factor of the fastest algorithm, which is here k-NN. Performances are cross-validated and averaged over the values of σ. This ill-balanced dataset might be the reason of the poor result of SVMs. The random forests obtains the best performances and a reasonable training time. This ensemble method is indeed easily parallelized. As a consequence, the SVM and k-NN are not tested further for this experimental setting.

The results for the buzz classification with a random forest classifier are presented below. These results are cross-validated, and average results are reported. Table 3.4 the precision, the recall, and the F-1 of the random forest classifier. Several values of the threshold σ are tested so that various applicative scenario are covered. The threshold σ varies in $[500; 4500]$ with step increment of 500. The results are given for two labeling functions: `growth_label` and `absolute_label`. Twitter and bulletin-board-system are tested. According to these results, to predict if a term is going to buzz is practicable when the forecast horizon is two days for Twitter, and fourteen days in the bulletin-board-system. More precisely, it is easier when the a buzz is defined with the `absolute_label` function. In this configuration, the value of the threshold σ as little importance. Indeed, for Twitter, F-1 score varies from 0.852 to 0.912. For the bulletin-board-system the F-1 score varies from 0.974 to 0.956. Such good results are easily explained by the fact that highly popular terms are very likely to remain popular when they are buzz candidates. On the contrary, the results for the `growth_label` function are not as good. Even if the precision is above 0.82 no matter what is the value of σ, the recall score drops to 0.480 on average for Twitter and 0.763 for the bulletin-board-system. For twitter the buzz classifier would miss more than half of the buzz in average. The results are better on the bulletin-board-system than on Twitter. These better results might be due to the slightly more balanced training-set. Another explanation lies in the difference of activ-

Algorithm	Precision	Recall	F-1	Runtime
RF	0.692 ± 0.08	0.470 ± 0.02	0.554 ± 0.02	$\times 2.5$
SVM	0.354 ± 0.07	0.622 ± 0.11	0.437 ± 0.03	$\times 16$
k-NN	0.675 ± 0.09	0.392 ± 0.04	0.492 ± 0.03	$\times 1$

Table 3.3: Comparison of the performances of three different classifier based on 20% of the Twitter-data. The labeling function considered is the `growth_label` function.

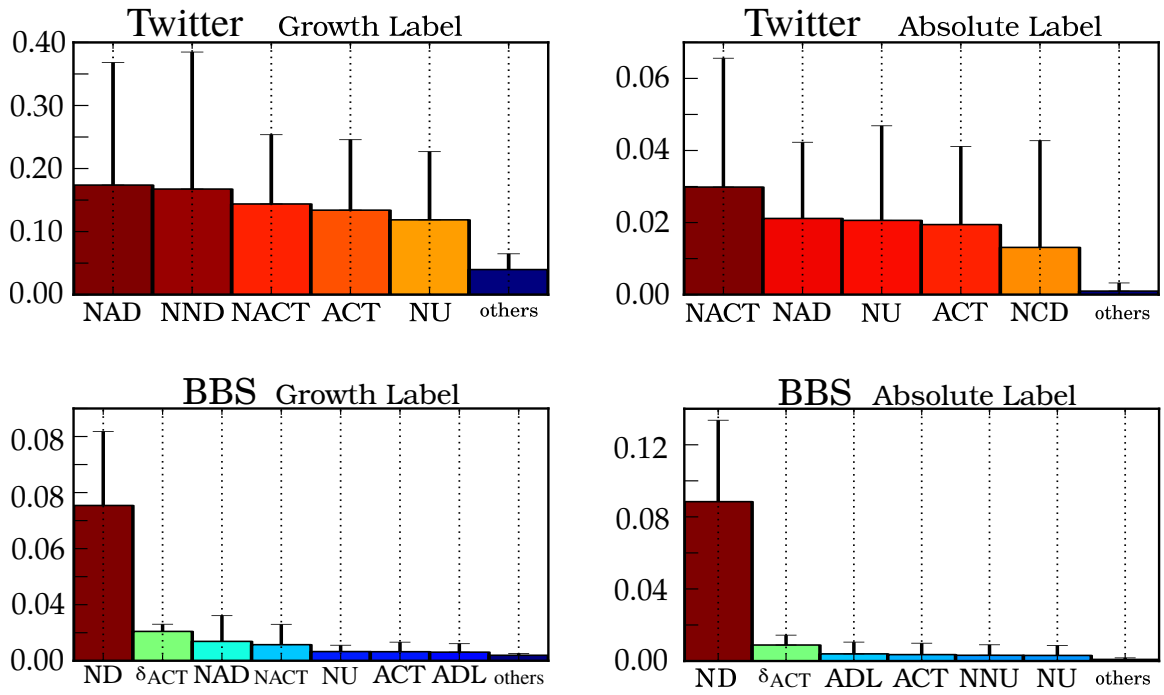

Figure 3.5: This figure presents the feature importances for the random forest classifier. These results are averaged over all the train-test splits. The average importance of the features that are not presented is named "others". The error bar represent the standard deviation. The bulletin-board-system is abbreviated BBS.

ity measure. Indeed, for the bulletin-board-system the activity measure, named ND, is more robust. This measure is not available in Twitter, for which the number of active discussions named NAD is used. The last line in Table 2.3.1 shows results averaged over all values of σ, when the single feature provided to the classifier is the activity measure. The activity measure is ND for the bulletin-board-system and NAD for Twitter. As presented in Table 3.4, this feature is sufficient to achieve very good classification scores on bulletin-board-system, for both labeling functions. The feature importance are used to complete this observation. Figure 3.5 presents the most important features according to the random forest classifier as described in Section 1.2.5. This figure makes clear that the proposed features are not used to predict the buzz in the bulletin-board-system. However, in Twitter five of the proposed features are used.

	absolute_label						growth_label					
	Twitter			bulletin-board-system			Twitter			bulletin-board-system		
σ	Prec.	Reca.	F-1	Prec.	Reca.	F-1	Prec.	Reca.	F-1	Prec.	Reca.	F-1
500	0.928	0.898	0.912	0.978	0.972	0.974	0.790	0.426	0.554	0.816	0.802	0.806
1000	0.910	0.872	0.890	0.970	0.960	0.966	0.855	0.535	0.657	0.830	0.780	0.804
1500	0.902	0.854	0.878	0.974	0.966	0.970	0.825	0.475	0.605	0.826	0.780	0.802
2000	0.884	0.830	0.858	0.968	0.968	0.966	0.815	0.455	0.585	0.838	0.798	0.818
2500	0.882	0.820	0.852	0.966	0.960	0.964	0.855	0.520	0.645	0.844	0.812	0.828
3000	0.888	0.818	0.852	0.966	0.952	0.960	0.825	0.485	0.605	0.836	0.762	0.796
3500	0.912	0.846	0.878	0.966	0.952	0.960	0.820	0.445	0.575	0.838	0.740	0.784
4000	0.920	0.870	0.894	0.956	0.958	0.958	0.795	0.465	0.585	0.838	0.716	0.770
4500	0.910	0.860	0.886	0.958	0.956	0.956	0.850	0.540	0.650	0.858	0.684	0.760
Avge	0.907	0.848	0.877	0.967	0.960	0.963	0.823	0.480	0.607	0.836	0.763	0.796
Activity only	0.893	0.840	0.884	0.969	0.962	0.962	0.704	0.409	0.510	0.858	0.809	0.831

Table 3.4: This table reports the precision, recall and F-1 score for Twitter and the bulletin-board-system. Two labeling functions are considered: `absolute_label` and `growth_label`. These labeling functions are test with various values of the threshold σ that controls the amount of activity in positive example.

3.3.2 Second experimental setting

The buzz classification is now tested on Twitter only, indeed it is the most difficult dataset. In addition, the French and German keywords are discarded. Indeed the volume of data collected with the French and German terms is small and less reliable than the data collected with the English terms. In order order to avoid the effects of ambiguity, 215 unambiguous terms are used. The unambiguous terms are those identified in Section 3.2.3 within the LOW list of terms. In order lower the risk of structural change in the collected data, 10 weeks of data-collection are used out of the 51 collected weeks. Indeed, the results of the Twitter REST API might change unexpectedly. Seven time-periods are used to build the feature matrices $\boldsymbol{X}$ and the label vectors $\boldsymbol{Y}$. Hence, the forecast horizon is equal to a week. For this experiment, the activity measure is the feature ACT. Indeed, this measure allows a more straightforward interpretation of activity in social-media. This measure is equal the amount of user-generated-content per term and time-period. The features presented in Section 2.3.1 are used in this experimental setting. The computation of the temporal correlation requiers to choose two parameters:

the rank r of the correlation, and τ the total amount of historical data to consider. The parameter τ is set to 28, hence the past three weeks of history are used to compute the temporal correlation. This value is chosen with respect to the amount a available data in this experimental setting. The rank parameter is set to $r = 7$. Indeed, Figure 3.6 makes clear that there exists weekly patterns. With the rank parameter $r = 7$, the temporal correlation is provided enough information to take into account the weekly pattern, if needed. Finally, the selection of buzz candidate, presented in Section 3.2.2, is no longer used.

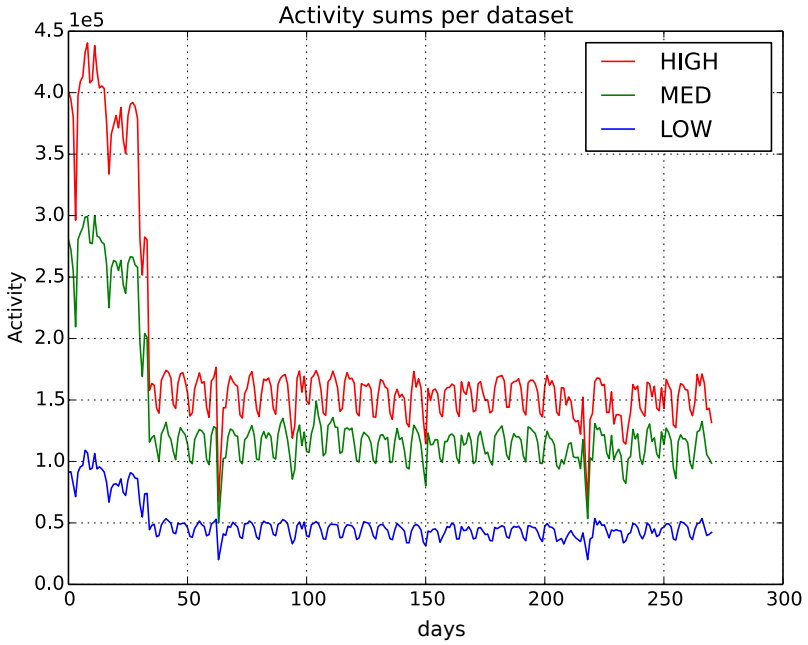

Figure 3.6: This figure presents the activity per lists of terms: LOW, MED, and HIGH presented in Section 3.2.3. The large drop around day 30 corresponds in a change of the Twitter-API. After day 50 a week-long pattern is clearly observable for the three list of terms. This figure shows also the two days: around day 65 and 225 for which the failure monitoring was triggered, as presented in Section 3.1.2.

The results of the buzz classification are presented for the `relative_label` labeling function. In this labeling function, the threshold σ controls the minimal growth factor for which an example is considered as positive. For instance with $\sigma = 2$, a term which activity doubles is considered as a buzz. Several values for the threshold σ are tested: $\{2, 3, 4, 5\}$. The training-set is ill-balanced for all of these values. There are never more than 1

positive example out of 10 negative examples, no matter the threshold value. Table 3.5 gives the number of positive and negative example for each value of the threshold σ. In addition, Figure 3.7 shows that the majority of terms that yield positive buzz example are feebly active during the week before the buzz. More precisely, most of the terms that yield a positive example had less than 200 contributions during the week of historical data. Consequently, the results of this experience might be valid solely for feebly active terms. In this test a SVM and a random forest are used. A grid search is used to select

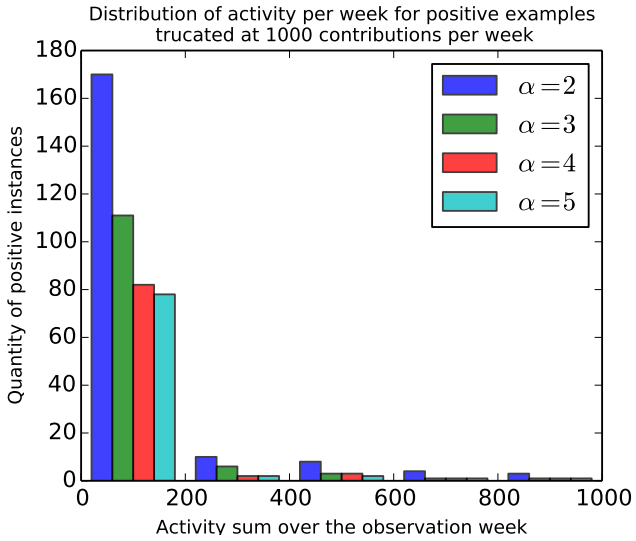

Figure 3.7: Distribution of the activity per term during the week before a Buzz. The threshold σ that controls the labeling function varies in $\{2, 3, 4, 5\}$. The labeling function is `relative_label`.

.

the meta-parameters of the classifiers. The grid search is identical to the one used in the previous experimental setting. Table 3.6 presents the detailed results for the two classifiers. The better prediction are obtained for the highest values of α, when one considers the F1-score. The F1-score varies from 0.47 (for $\alpha = 2$) to 0.80 (for $\alpha = 5$) with the random forest classifier. In this setting, the highest growths are predicted with greater confidence. This is a surprising results given the balance of the training-set. However, an sensible explanation could be that some of the terms already begun to buzz during the week used to compute the features. The output of random forests is used to estimate the feature importances. As the importance of a feature varies for each train-test split we report here the average importance over each split. The most important feature is the user engagement evolution, abbreviated δ_{UEL} as presented in

Section 2.3.1. More precisely, the most important feature is δ_{UEL} when observed seven days before the buzz. There are no distinguishable patterns in the other most discriminative features. Among the features that take into account the correlation between keywords the Minimum Weighted Correlation and the Maximum Weighted Correlation are the most discriminative, although, theirs importances are low.

σ	*#Negative*	*#Positive*	Ratio
2	1943	207	0.107
3	2026	124	0.061
4	2054	96	0.047
5	2062	88	0.043

Table 3.5: Balance of the classes for the buzz classification with respect to the threshold parameter σ. The ratio given is number of positive examples out of negative examples. With $\sigma = 2$, the `relative_label` function labels as positive the example whose activity doubled or more.

	$\alpha = 2$			$\alpha = 3$			$\alpha = 4$			$\alpha = 5$		
	Prec.	*Recall.*	F1	*Prec.*	*Recall.*	F1	*Prec.*	*Recall.*	F1	*Prec.*	*Recall.*	F1
RFC	0.74	0.35	0.47	0.98	0.51	0.67	0.99	0.68	0.80	0.99	0.73	0.83
SVC	0.28	0.48	0.33	0.82	0.49	0.55	0.90	0.67	0.76	0.95	0.72	0.81

Table 3.6: Detailed results of the buzz classification for four thresholds $\alpha \in \{2, 3, 4, 5\}$ using a random forest classifier (abbreviated RFC) and a support vector classifier (abbreviated SVC). The scores: precision (abbreviated *Prec.*), *Recall*, and F1 are given only for the positive class since the number of positive example does not allows to use a weighted average over positive and negative classes.

The experiments on the buzz classification give several insights. For the bulletin-board-system, to predict a buzz is feasible. Despite the various features that are proposed, none is more important than the target feature it self. It is hard to assess if this observation can be generalized to other bulletin-board-system than those observed here. In Twitter to predict a dramatic increase of activity is more difficult. Three labeling

functions are tested to qualify the activity increase. However, theirs results cannot be compared because two different experimental settings were used. For Twitter, it is easy to predict the terms for which the activity will be greater than a fixed threshold. This result is not surprising, as popular term are likely to remain popular in the next couple days. To predict if the activity of a term will increase more than a fixed amount is also practicable, yet less easy. To predict if the activity of term will double is also a difficult task. Globally, the precision in these task are acceptable. Hence the outputs of a buzz classifier on Twitter might be sufficiently reliable. However the recall is globally low, and numerous buzz would be missed by these classifiers.

3.4 Magnitude prediction

The section is devoted to the evaluation the magnitude prediction problem. This problem is tackled with state-of-the-art machine learning methods and the features presented in Section 2.3. In the magnitude prediction the task is to predict, for each term, the total activity during the labeling. This task corresponds to a regression as proposed by Kawala *et al.* [86]. The magnitude prediction problem is studied with the two experimental settings presented in the previous section. In the first experimental setting, terms are filtered to keep those with an upward tendency in their activity. On the contrary on the second experimental setting, terms are filtered to keep those which are the most meaningful. More generally, the second experimental setting aims at have a more controlled environment and reproducible experiments. The parameters used for the extraction of examples are identical to those presented in Section 3.3. The activity measures are also unchanged.

3.4.1 First experimental setting

For Twitter seven time-periods are used to build the matrices of features named $\boldsymbol{X}$, two time-periods are used to build the label vectors named $\boldsymbol{Y}$. For the bulletin-board-system, the matrices of features are based on sixty time-periods, and the label vectors on fourteen time-periods. A time-period is equal to one day. The random forest provides efficient classifiers for the buzz prediction task, hence a random forest-Regressor is also used for the magnitude prediction task. A grid search is used to select the meta-parameters of the regressor. The results are cross-validated and the evaluation measures are averaged over the different train-test splits. The coefficient of determination, or R^2 is broadly used to

measure the performances of a regression. The coefficient of determination measures the prediction-error and takes into account how scatter the examples are. The highest value of R^2 is 1 when the prediction is prefect. The readers interested in more details about R^2 are referred to the book by Everitt [53]. For the bulletin-board-system, the coefficient of determination is 0.972, for Twitter it is 0.942. These values suggests that activity can be accurately predicted within these settings. However, the coefficient of determination might be too sensitive to very scattered values. The data-collection is made with respect to terms of heterogeneous popularity, hence an additional analysis is needed. This analysis is based on the normalized prediction error. A term whose activity is Y and prediction is $\tilde{Y}$, has a normalized error of $|\tilde{Y} - Y|/Y$. For instance, when the value to predict is 500 and the predicted value is 750, then the normalized prediction error is 0.5. Figure 3.8 presents the histogram of the normalized error for the examples from Twitter that had a significant activity increase. This figure shows that 20% of the examples are predicted with a normalized error lower that 4%, and 74% of the examples are predicted with a normalized error lower that 23%.

The feature importance for this regressor is studied. Figure 3.9 presents the most important features. All the most important features, no matter the social-media, are closely related to the activity measure. For the bulletin-board-system the single most used feature is the activity measure. For twitter the most important feature is ACT. This feature is used as activity measure in the second experimental setting.

3.4.2 Second experimental setting

In this experimental setting the magnitude prediction is tested with the data collected from Twitter, a restricted set of terms, and the ACT feature as activity measure. Two regressors are tested. The first one is based on the support vector machine, abbreviated SVR. The second regressor is based on random forest, abbreviated RFR. A grid search is used to select the meta-parameters of these regressors. The forecast horizon is set to seven time-periods. Hence the task is here to predict the activity of a term during the upcoming week with respect to the past week. The objective is to provide actionable insights on the magnitude prediction. The mean absolute error, abbreviated MAE, is used for these tests. Indeed the interpretation of the MAE is very straightforward. The MAE is equal to the average of the absolute values of the prediction errors. Hence the MAE does not distinguish positive errors from negatives ones. The MAE might be less

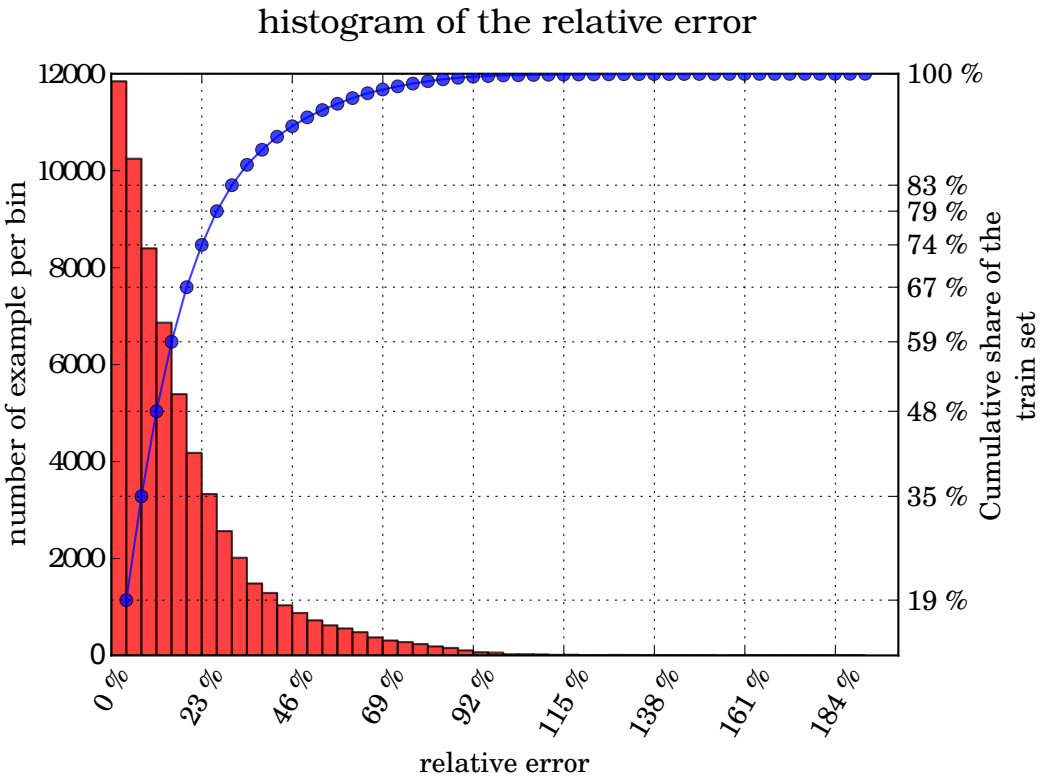

Figure 3.8: This figure presents the normalized error binned such that each bin represent approximatively an increase of 4% in normalized error. The blue dotted curve presents for each of these bins the cumulative amount of test-examples. Therefore, the sixth leftmost dot on the blue line indicates that the six first bins contains 74% of the examples. On the abscissa we read that the highest normalized error is 23% the six first bins. These are examples from Twitter.

interpretable when the values of the labels are scattered. The popularity of the observed terms is not uniform, thus their labels are likely to be scattered. Therefore, the activity levels are discretized in several non-overlapping intervals and the MAE is computed for each of them independently. For instance, the error for a term that has 60 `contents` during the period of observation is reported in the interval $[0, 64[$. On the contrary the error for a term that has 128 `contents` is reported in the rank $[64, 256[$, and so forth. In Table 3.7 the R^2 coefficient indicates that the overall regression is correct, while the MAE on each interval shows that errors in regression are acceptable for terms that are highly active. On the contrary the error on the feebly active terms is relatively high.

The instability of term is studied in order to gain more insights on the usability of

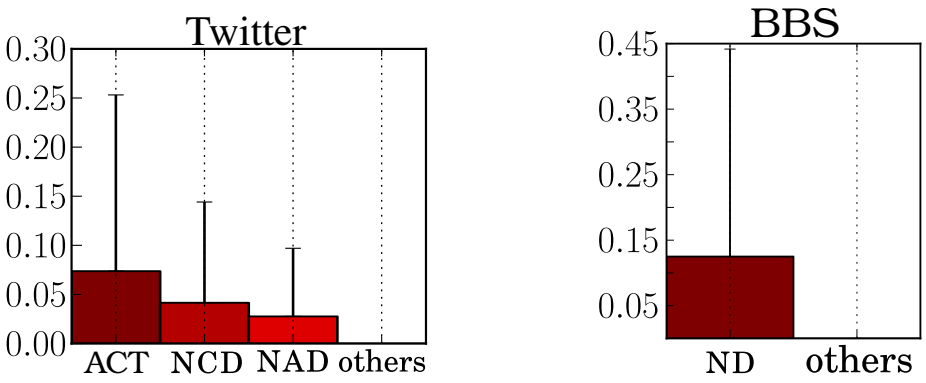

Figure 3.9: This figure presents the feature importances for the random forest-Regressor. The feature importances for Twitter appears on the left side of this figure. On the right side the feature importances for the bulletin-board-system are presented. These results are averaged over all the train-test splits. The average importance of the features that are not presented is named "others". The error bar represent the standard deviation.

the magnitude prediction. An instable term alternates quickly high activity levels and low activity levels. The more instable a terms is, the harder it should be to predict its activity. The instability of a term is the average of ratio between its activity over a week and its activity over the next week. Hence the lower bound of the instability is 0, and it has no upper bound. With the observations of this experimental setting, the normalized mean absolute error of RFR is correlated to the instability level of a term. Indeed, The Pearson correlation coefficient of these two measures is 0.717 when values above the 99^{th} percentile of the mean normalized are discarded. SVR produces more outliers. The Pearson correlation coefficient is 0.598 when values above the 80^{th} percentile of the mean normalized are discarded. The two-tailed *p-value* is null for RFR and SVR. According to these observations, our proposal for the magnitude prediction should be used for keywords that are either highly active or sufficiently stable through time. Figure 3.10 presents the instability levels with respect to the normalized mean absolute error values obtained with RFR and SVR.

The experiments on the magnitude prediction give several insights. For the bulletin-board-system, to predict the activity is feasible. The target feature is the only one to have a real importance for the random forest regressor. Hence, in this case, the proposed formalization is of no use. In Twitter to predict the activity is harder, no matter the activity measure. Nonetheless, the proposed approach allows to predict with reasonable error the activity for different forecast horizons. However, the most instable terms and

the feebly popular terms are hard to predict. The typical error for those term might prevent an industrial use of the proposed approach. The future activity of terms that have more than 4096 `contents` per week is the most accurately predicted.

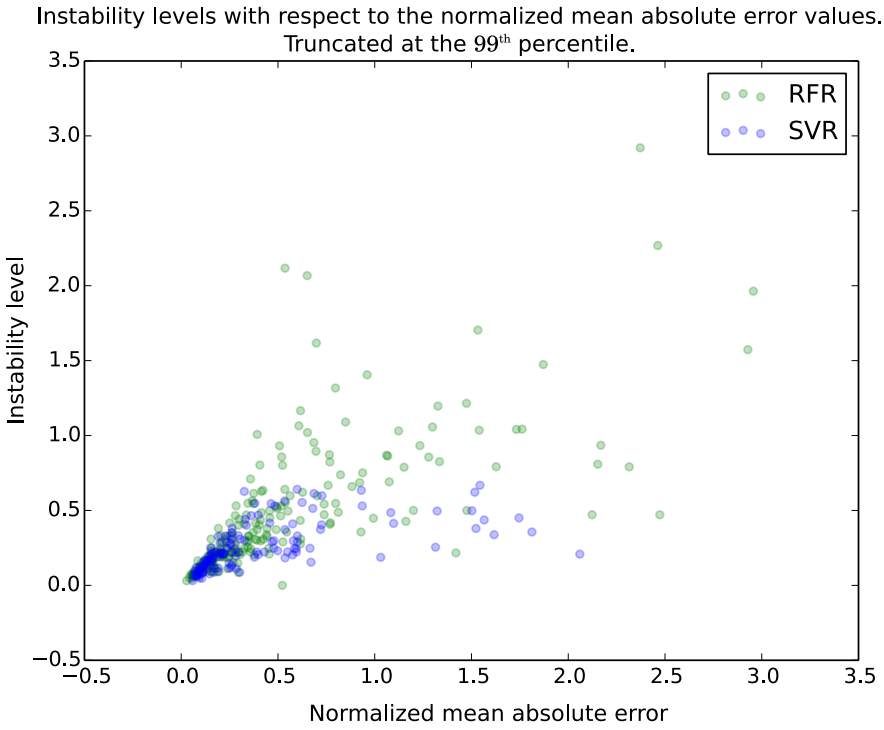

Figure 3.10: This figure presents a scatter plot of the instability level with respect to the normalized mean absolute error. This figure illustrate the correlation between instability level of the terms and the normalized mean absolute error of a random forest regressor (abbr. RFR) presented in blue, and a suport vector regressor (abbr. SVR) presented in green. The Pearson correlation coefficient is 0.716869 and the two tailed *p-value* is null.

3.5 Rank prediction

This section studies the rank prediction problem. In rank prediction problem one has to learn a ranking function that sort the terms accordingly to their upcoming activity levels. This problem is tackled with two standard learning-to-rank approaches: point-wise and pair-wise, as proposed by Kawala *et al.* [87]. The point-wise approach is closely related to the magnitude prediction presented in Section 3.4. Indeed, in this case, the ranking function to be learned is a regression on the rank of each term, and the rank is defined

	R^2	MAE					
		< 64	< 256	< 1024	< 2048	< 4096	≥ 4096
RFR	0.9669	17.5	69.5	141.6	402.4	569.9	764.0
SVR	0.9665	31.2	55.5	136.6	352.5	554.5	764.5

Table 3.7: The coefficient of determination R^2 and the mean absolute error (abbreviated MAE) for each regressors: random rorest regressor (abbreviated RFR), and support vector regressor (abbreviated SVR). The activity intervals do not overlap.

by the activities of terms. The pair-wise approach considers each pair of terms in order to decide which of the two terms will have the higher rank. The term which was ranked higher the most times obtains the first rank, and so forth. The history and forecast horizon are based on seven time-periods of one day and the activity measure is the ACT feature.

3.5.1 Evaluation measures for learning-to-rank tasks

The performances of the ranking functions are evaluated with two standard measures. The first measure is the Kendall tau rank correlation coefficient. This coefficient varies within $[-1, 1]$. When this coefficient is null, it indicates that the ranking function and the actual ranking are not correlated. When this coefficient is 1, it indicates that the ranking function and the actual ranking agreed for the ranking of every pair of terms. On the contrary when this coefficient is -1, the ranking function and the actual ranking disagreed for the ranking of every pair of terms. The second measure is the Spearman coefficient of rank correlation. This coefficient varies within $[-1, 1]$, and takes into account the difference between the predicted rank and the actual rank. The readers interested in more details about these two measures are referred to the books by Hollander *et al.* and Kendall [78, 89]. An additional coefficient is proposed to evaluate the utility of a ranking model for industrial applicative constraints. This coefficient measure the capacity of a ranking model to predict sudden changes in the top-n ranks. Indeed it is highly valuable to determine which terms that are not yet among the top-n most active terms will belong to the top-n most active term is the future. Such terms are designated as "newcomers". The proposed coefficient is named the surprise coefficient, it is defined as follows. For a set of topics $\mathcal{Z}$, a ranking r, and an observation $\boldsymbol{Y}_{\mathcal{Z}}$ of the topics in $\mathcal{Z}$, we have $top_n(r, \mathcal{Z}) = \{e \in \boldsymbol{Y}_{\mathcal{Z}} : r(e) < n\}$. Newcomers are then defined with respect to the

ranking during observation that is named r_{obs}. The definition of the newcomers is as follows.

$$\texttt{newcomers}(\mathcal{Z}, n) = \{e \ \in \boldsymbol{Y}_{\mathcal{Z}} \ : e \in top_n(r_{true}, \mathcal{Z}) \wedge e \notin top_n(r_{obs}, \mathcal{Z})\}$$

In this equation r_{true} is the actual rank that is observed at evaluation period. This definition of `newcomers` and the predicted rank for the evaluation period named r_{pred} allows to define the n-`surprise` score as follows:

$$\texttt{surprise}(\mathcal{Z}, n) = \frac{|top_n(r_{pred}, \mathcal{Z}) \cap \texttt{newcomers}(\mathcal{Z}, n)|}{|\texttt{newcomers}(\mathcal{Z}, n)|}$$

The `surprise` takes values in $[0, 1]$, where 0 indicates that the learning model predict none of the newcomers in the top-n and thus do not cope with sudden changes in the ranking. On the contrary a `surprise` score equal to 1 indicates that sudden changes are perfectly predicted. The parameter n of the `surprise` score is set with respect to application needs. Here the parameter n is set to 30, indeed it seems to be a good compromise for a human operator that would have to perform actions on each term of the top-n.

3.5.2 Effects of ambiguity and activity

The rank prediction is evaluated with respect to the three lists of terms presented in Section 3.2.3. With these three lists, the terms are grouped with respect to their ambiguity. The rank prediction is tested on each list of term so that it is possible to assess of the ambiguity effect for the rank prediction. The rank prediction is tested with four different learning-to-rank methods and a naive baseline. Firstly, a pair-wise approach with SVM-Rank. In this approach a support vector machine classifier is used to ranks the pairs of terms. Secondly, a point-wise approach that is based on a random forest regressor. Two other point-wise approaches are tested, one is based on a gradient boosted tree, the other is based on a support vector regressor. The baseline is straightforward, it consists in ranking the terms according to their rank at the end of the time-period of observation. The tests are cross-validated and average results over all train-test splits are reported.

Figure 3.11 shows the mean score per dataset and learning model. The error lines presents the standard deviation over all the train-test splits. On the top left part, the Spearman rank correlation coefficient indicates that the pair-wise method performs equally or better than the three point-wise methods. In addition, the pair-wise has more

stable results across the different folds. The baseline obtains results almost as good as the three learning-to-rank methods. However, the baseline is highly unstable as shown by the error lines. The ambiguity has a direct effect on the capacity of the trained rankers to predict correctly the future rank. Indeed, the results of all the learning-to-rank methods get worst and less stable when more ambiguous terms are considered. The top right part of Figure 3.11 presents comparable results for the Kendall tau rank correlation coefficient. The bottom part of this figure presents the surprise score. The baseline has a null surprise score as per the its definition. Indeed the baseline reproduces the last observed ranking, and the surprise score measure the ability to predict newcomers in the top most ranks. The point-wise method obtains a mean surprise score of 0.38 while the pair-wise method obtains 0.40. Hence, in a practical situation these methods would predict less than half of the terms that will become one of the thirty most active terms. The rank prediction in not well suited to detect terms that are about to grow enough to be among the top-30. The ambiguity of the terms has less effect on the surprise score than it has on the Kendall tau rank correlation coefficient and the Spearman rank correlation coefficient.

Discretized heat-maps are used to have more insights on the effects of ambiguity of a term on its ranking error. In a discretized heat-maps the predicted ranks are presented on the ordinate axis and the actual ranks are presented on the abscissa axis. The rank are binned per ten and the lowest ranks are on the rightmost part of the illustration, hence the prediction error for the top-10 rank is presented on the leftmost column. In this illustration a prefect ranking is presented a the identity line. Figure 3.12 present four discretized heat-maps that are colorized with respect to the ambiguity of the terms. There is not clear pattern that relates the ambiguity to the ranking-error.

Discretized heat-maps are also used in order to have more insights on the effects of the activity of terms on its ranking error. Figures 3.13, 3.14 and 3.15 presents discretized heat-maps that are colorized with respect to the activity of the terms to be ranked. Figure 3.13 reveals that ranking error of the baseline is more scattered than for point-wise or pair-wise methods. Figures 3.14 and 3.15 shows that the ranking error is the lowest below rank 50 and increases proportionally to the rank. Above rank 200, typical errors are to high for the ranking to be used. These patterns are reproducible for the two lists of terms and the two tested methods.

Finally, the stability of the rank prediction is tested with respect to the number of

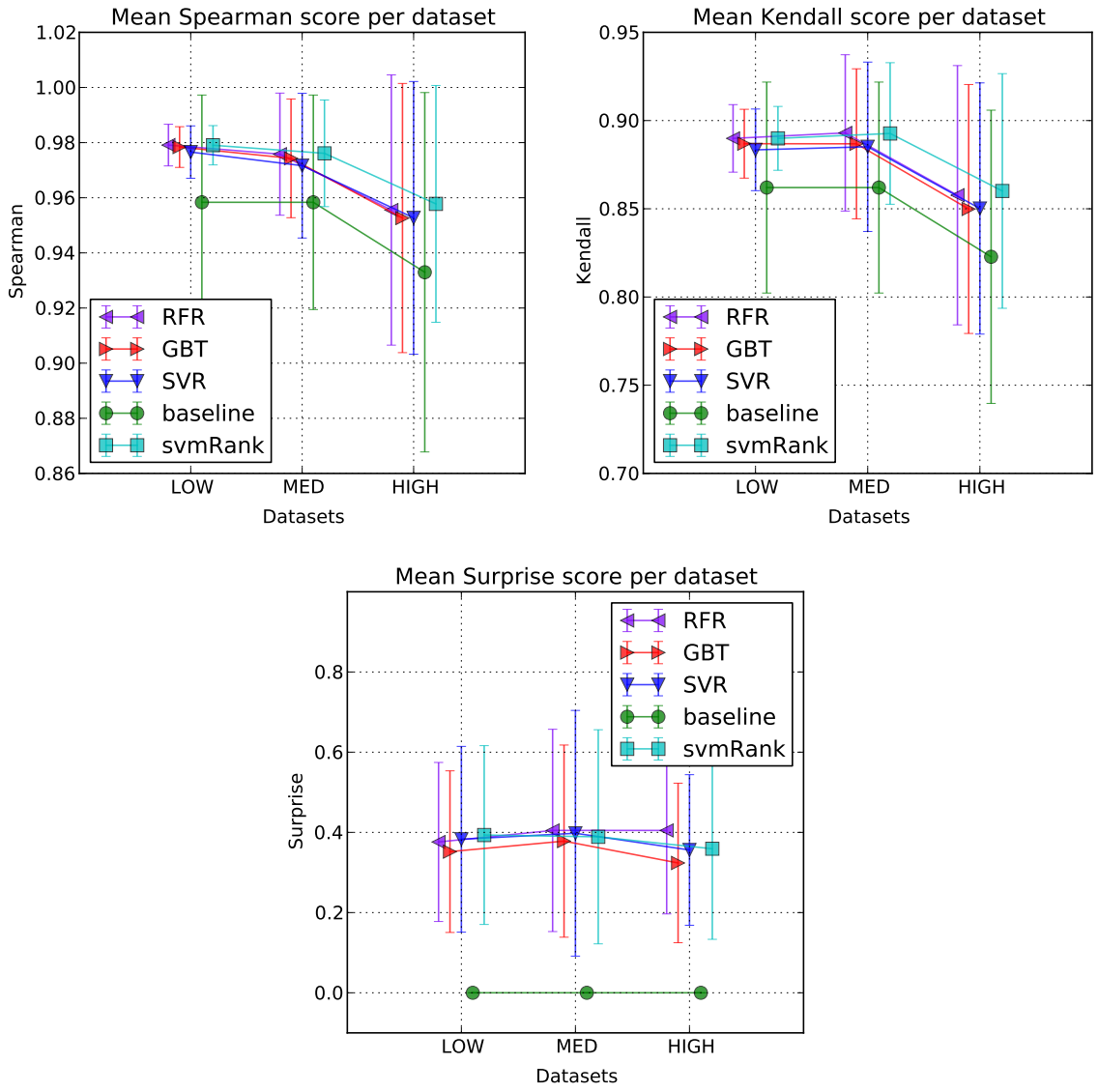

Figure 3.11: This figure presents three evaluations of the ranking error for several learning-to-rank methods. These results are averaged over the different train-test splits, and the error bars presents the standard deviation. The Spearman rank correlation coefficient is presented on the top-left side. The Kendall tau rank correlation coefficient is presented on the top-right side. The surprise score is presented on the bottom line. Three point-wise methods are tested they are abbreviated RFR for the random forest regressor, SVR for the support vector regressor, and GBT for the gradient boosted tree. One pair-wise method is tested, it is abbreviated SVMRANK.

terms to be ranked. A point-wise method based on the gradient boosted tree is used. The number of terms to be ranked is iteratively incremented from 100 to 1300 with a step size of 100 terms. For each iteration a new model is trained and the Spearman rank correlation coefficient is measured. This experience showed that this approach has stable results with respect to the number of terms to be ranked. Indeed, the Spearman rank correlation coefficient is stable, with 0.9737 for 100 terms, 0.9781 for 600 terms, and 0.9765 for 1300 terms.

The experiments on the rank prediction were done with data collected in Twitter and three sets of terms classified with respect to their ambiguity. These experiments gives several insights on the usability of the ranking prediction. Firstly, as excepted, the ranking error worsen when ambiguity increases. Indeed, experiments based on the lists of terms MED and HIGH which contains ambiguous terms obtains worst and less stable results. However a finer analysis on the average ranking error with respect to the ambiguity of terms to be ranked did not revealed any clear error pattern. Secondly, the most active terms are the easiest to rank. More precisely, the error in top-50 is acceptable and growth such that it is pointless to use the prediction for the ranks beyond 200. Finally the capacity for the ranker to predict newcomers in the top-30 is tested. Less than half of the newcomers would be detected with the proposed methods.

Conclusion

This chapter presents experiments related to three activity prediction problems. A significant effort was necessary to collect data at a large scale during a year. The different steps of preparation for the collected data are presented in this chapter. The three activity problems presented in Section 2.2 are studied. More precisely the features proposed in Chapter 2 are used to train supervised machine learning algorithms. The ability to predict the future activity is tested with respect to two different experimental settings. The first experimental setting allows to compare the activity prediction on Twitter and a bulletin-board-system. The second experimental is dedicated to twitter only. The experiments show that the prediction is feasible for the three activity prediction problems. However a cautious study of the results reveals that there are several settings for which the prediction fails to be accurate.

Regarding the bulletin-board-system, the experiments show that the buzz classifica-

tion and the magnitude prediction are accurately predicted. More precisely, the most important feature is the target feature itself. Here the target feature is the number of page-views. It is hard to assess if this observation can be generalized to many bulletin-board-system. Indeed, we do cannot know the the number of page-views for other bulletin-board-systems. In this case, the proposed formalization is of no use.

Regarding Twitter, the experiments show that it is harder to predict the future activity of a term. Three labeling functions are proposed for the buzz classification. These three labeling function cover different applicative scenarios, and the error in prediction varies accordingly. The `absolute_label` function, defined in Equation 2.3, obtain the better results. Indeed, this function labels each term with respect to a fixed activity threshold, and highly popular terms are likely to remain popular in the near future. The `growth_label` and the `relative_label` describe more difficult applicative scenarios. Indeed, for these two functions, the buzz is measured as an increase of activity. Although the precision obtained for these two labeling functions is acceptable, the recall is very low, especially for the moderate growth thresholds. Hence, the trained classifiers would miss an important share of the positive examples, presumably, the keyword which had the most sudden activity growth. The magnitude prediction is also tested against Twitter-data. Experiments shows that this regression fails to predict the activity of the most instable terms as well as the feebly popular terms. The typical error for those term might prevent an industrial use of the proposed approach. On the contrary the prediction obtains better results with the terms that have more than 4096 `contents` per week. The experiments on the rank prediction were done with three sets of terms classified with respect to their ambiguity. As excepted, the average ranking error worsen when ambiguity increases. Indeed, experiments based on the lists of ambiguous terms MED and HIGH obtains worst and less stable results. A finer analysis on the average ranking error of terms in the MED and HIGH lists did not revealed any clear error pattern that would relate the ambiguity of a term to the ranking error. Indeed, terms with the highest and lowest ambiguity might obtain arbitrary high or low ranking error. On the contrary terms with mean ambiguity are less prone the extreme ranking error.

These experiments outlines that the generic framework, and the feature based on it, are able to capture the most obvious activity change, no matter the chosen activity prediction problem or social-media. However, for the most difficult cases, as for example a sudden change in activity, relevant features are yet to be defined.

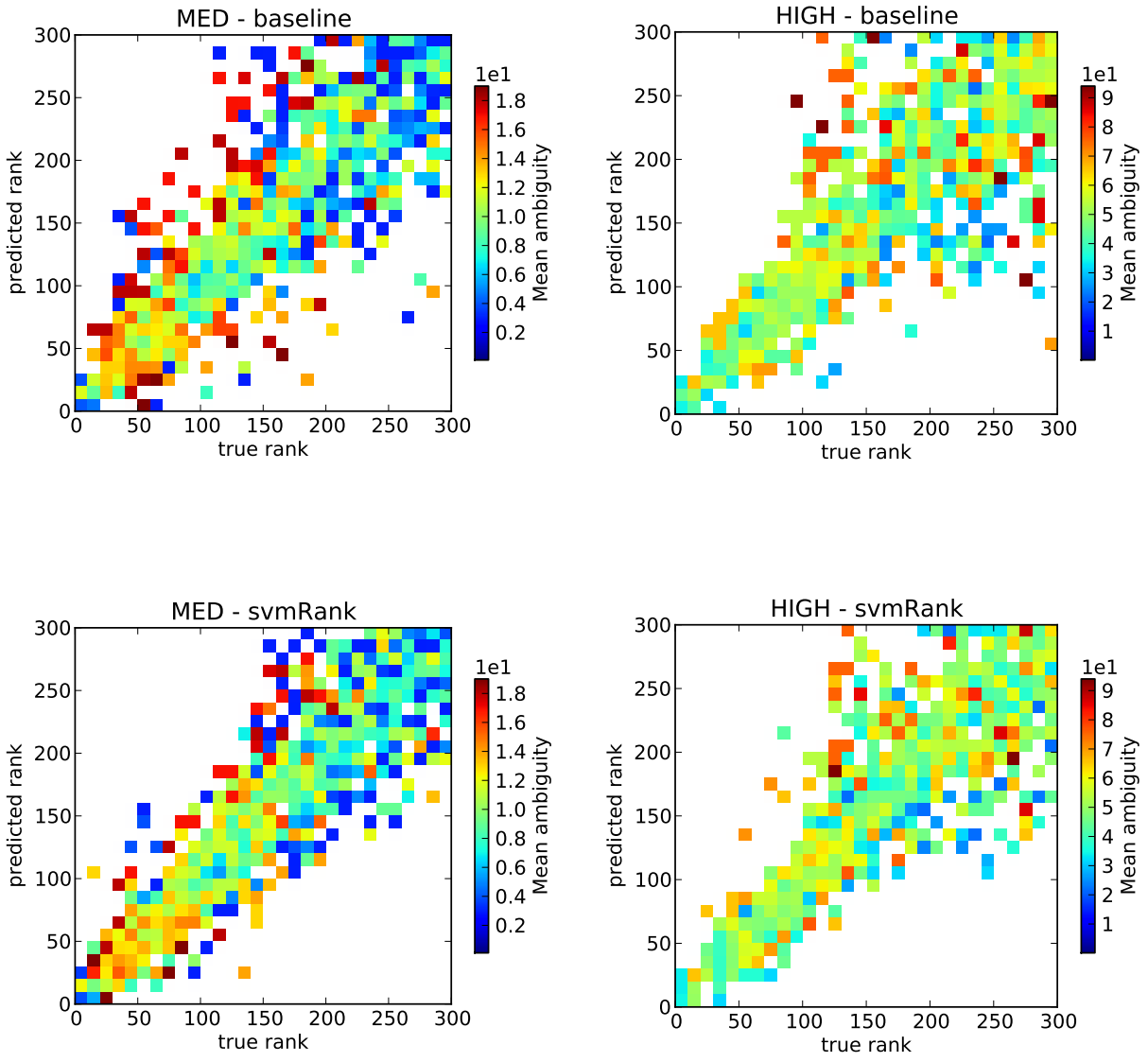

Figure 3.12: The ranking error is presented with respect to the term-ambiguity. The results are presented for the terms in the MED and HIGH lists. The ranking error is binned by 10, hence the bottom-left-most square presents the correct ranking for ranks one to ten. The top line presents the ranking error of a naive baseline that repeats the ranks observed at the end of the observation period. The bottom line presented the ranking error of a pair-wise method with support vector classifier. The left side of this figure presents the ranking error for terms in the MED list, and the right presents the ranking error for the terms in the HIGH list.

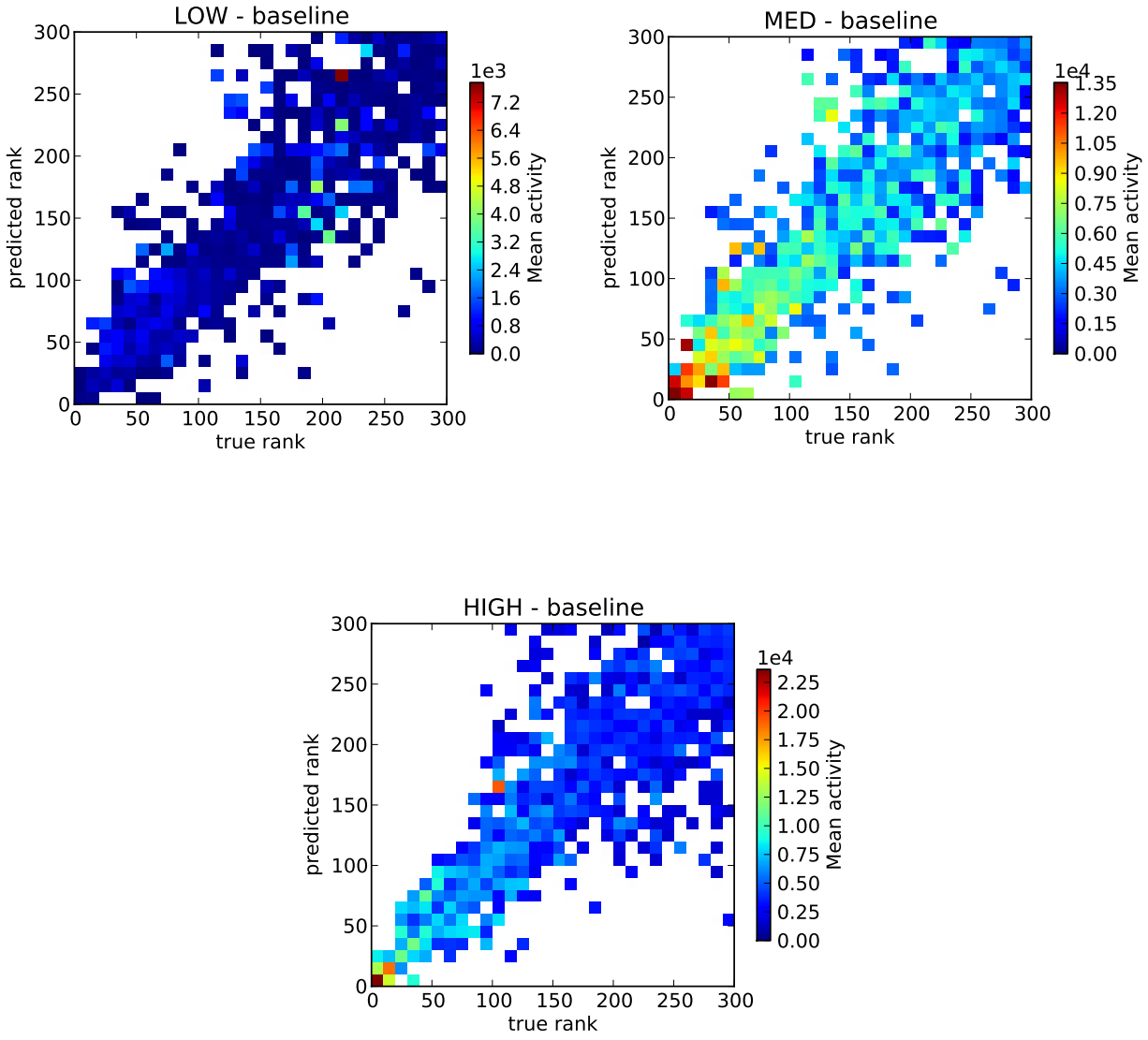

Figure 3.13: The ranking error is presented with respect to the activity level during the observation period. The ranking method is a naive baseline that repeats the ranks observed at the end of the observation period. The results for the terms of the LOW and MED lists are presented top line. The results for the terms of the HIGH list are presented on bottom line. The ranking error is binned by 10, hence the bottom-left-most square presents the correct ranking for ranks one to ten.

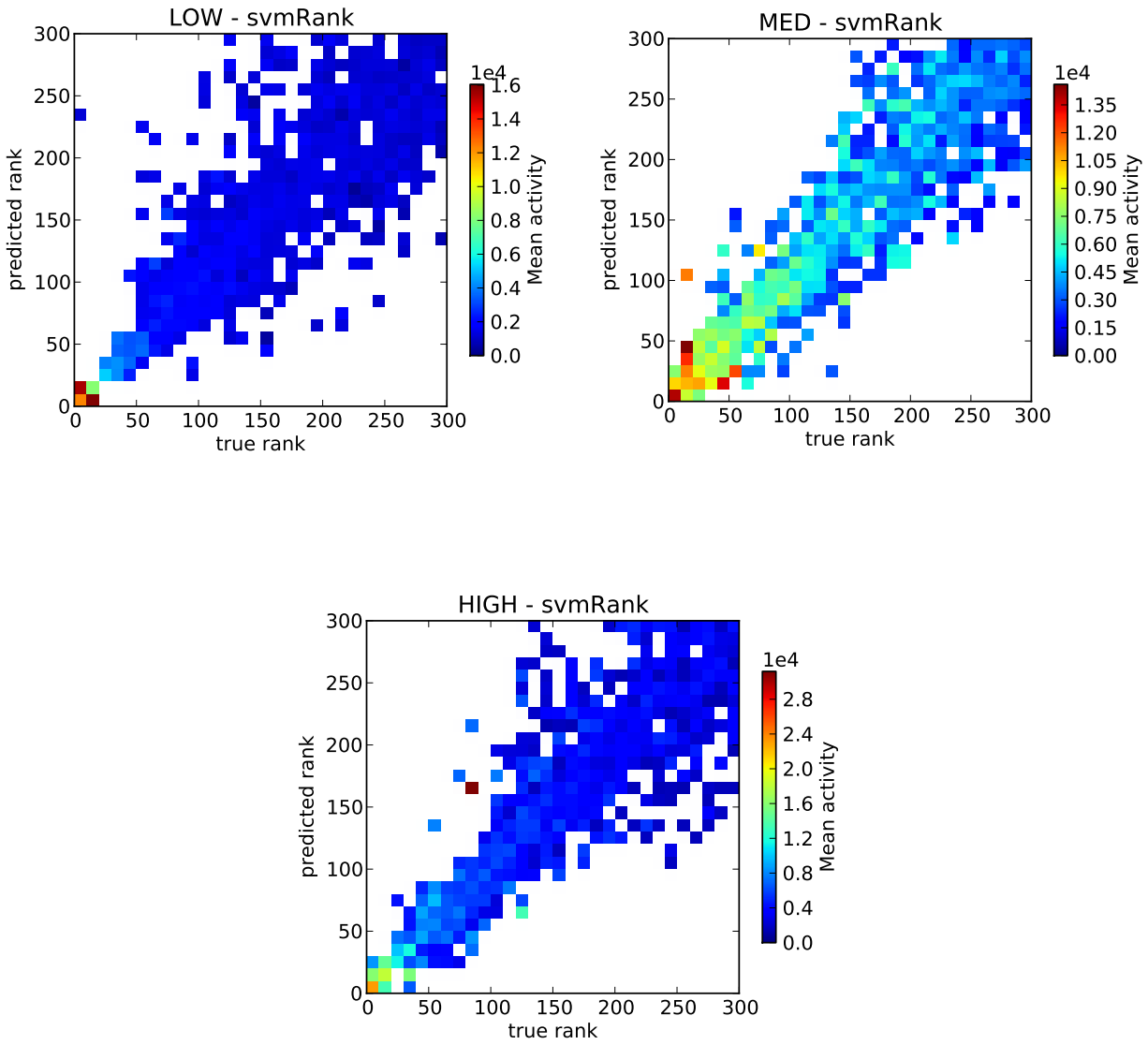

Figure 3.14: The ranking error is presented with respect to the activity level during the observation period. The learning-to-rank method is pair-wise with support vector classifier. The experiment is cross-validated and average results are presented. With this representation a perfect ranking produces the identity line. The results for the terms of the LOW and MED lists are presented top line. The results for the terms of the HIGH list are presented on bottom line. The ranking error is binned by 10, hence the bottom-left-most square presents the correct ranking for ranks one to ten.

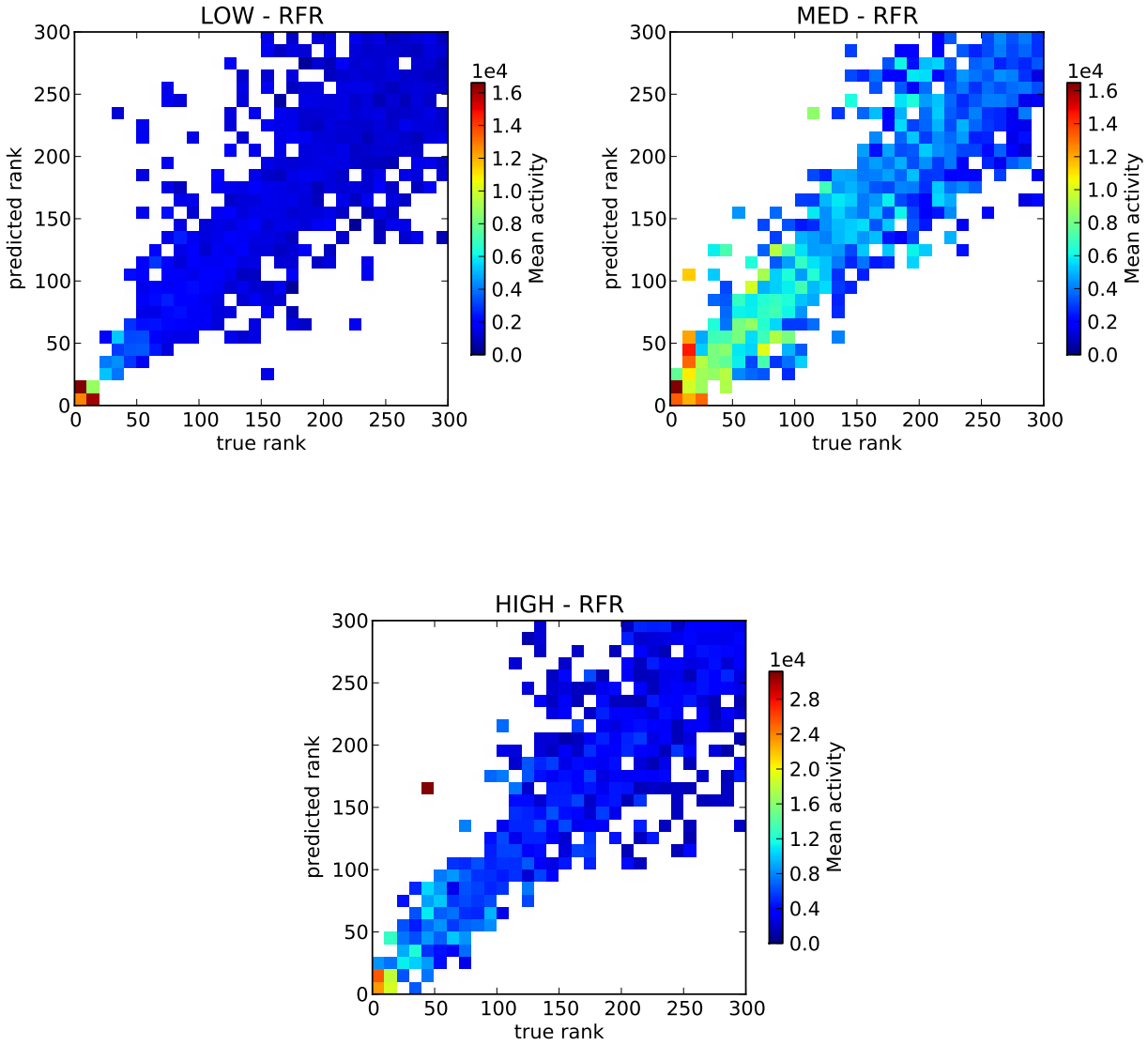

Figure 3.15: The ranking error is presented with respect to the activity level during the observation period. The learning-to-rank method is point-wise with a random forest regressor. The experiment is cross-validated and average results are presented. With this representation a perfect ranking produces the identity line. The results for the terms of the LOW and MED lists are presented top line. The results for the terms of the HIGH list are presented on bottom line. The ranking error is binned by 10, hence the bottom-left-most square presents the correct ranking for ranks one to ten.

Conclusions and future work

This study is dedicated to three different activity prediction problems that cover different applicative needs. In these problems the activity is measured for topics such as "smartphones" or "laptop", as presented in Chapter 1. These activity prediction problems are defined with a generic framework that is presented in Chapter 2. That framework allows to define a *social-media-mining* problem independently of an actual social-media. Hence it is easier to adapt a solution to several social-media. This requirement was driven by the envisaged industrial applications of this study. More generally, this approach is sensible when one considers the increasing number of different social-media. The activity prediction problems are tackled with supervised machine learning algorithms. Numerous experiments are realized to determine which settings yields the best prediction of the activity. Chapter 3 presents these experiments.

Chapter 1 introduces the *social-media-mining* from a practitioner point of view. Hence, this chapter lists several definition attempts of a social-media. In addition practical matters are discussed, as for instance: the automated collection of data from social-media; or the principles of the machine learning. The state of the art on the activity prediction in social-media presents three research axes: firstly the spikes of collective attention, secondly topic popularity, and thirdly the user-generated-content popularity. These studies belongs to the second research axis.

Chapter 2 introduces a generic framework that describes various social-media. In this chapter three activity prediction problems are presented. Each of these definitions is tailored for specific applicative requirements. The magnitude prediction is a regression problem that aims to predict the activity that will be observed in a social-media. The buzz classification is a binary classification problem that aims to predict which topics will have a burst in activity in the near future. The rank prediction is a learning-to-rank problem that aims to predict the relative importance of each topic. In addition, features

based on the generic framework are defined in order to tackle these three activity prediction problems with supervised machine learning algorithms. The generic framework uses topic-models to group together user-generated-contents that are related to each others, but none particular topic model is required. In this study, the topic-model amounts to a simple look-up in a product-data base applied to each token of each user-generated-content. The user-network associated to a social-media is not described in the generic framework. Indeed, to keep an up-to-date version of the user-network is a non acceptable burden in the envisaged industrial applications of this study.

Chapter 3 presents the collection of the data and the different preparation steps. The data is collected in two social-media: Twitter the most widely used micro-blogging system; and a bulletin-board-system dedicated to high-tech subjects. The collection of data lasted 51 weeks and was shared publicly. Two experimental settings are defined and tested. The first experimental settings is used to compare the results on Twitter and the bulletin-board-system. This comparison is made on a long time-scale and with three different languages. The second experimental setting uses Twitter-data solely. In order to increase the reproducibility, this experimental setting is based on user-generated-contents collected with respect to a list of unambiguous English terms over a ten consecutive weeks. A cross-validation method that takes into account the time of observations is defined. Indeed, the problems that are studied are time dependent and classical cross-validation methods might be inaccurate. An unsupervised method to extract buzz candidates based on the past activity levels is also proposed. The influence of the ambiguity on the activity prediction is studied. Wikipedia is used to determine the ambiguity of keyword. Contrasted results are obtained. In the bulletin-board-system, the experiments show that the most of the predictive power lies in the target feature, hence the proposed features and formalization are not necessary. On the contrary, Twitter revealed more interesting results. The precision obtained for the labeling functions that describe a growth in activity is acceptable. However, the recall is very low, especially for the moderate growth thresholds. Hence, the trained classifiers would miss an important share of the positive examples. In this situation, the keywords which had the most sudden activity growth are not likely to be detected. The ambiguity influences the results of the rank-prediction, however the precise role of the ambiguity remains to be determined.

This study proposed a unified representation for social-media that is used to define three different activity prediction problems. Based on a large scale data collection on two

social-media, this study enlighten those situations which are the hardest to deal with when one aims to predict the future activity of a topic. This work opens research perspectives that are briefly summarized below. Firstly the unified representation for social-media could be extended so that it allows to take into account non textual user-generated-contents. Moreover, one could leverage partial knowledge of the user-network. Hence, one could extended the unified representation for social-media so that include partial knowledge of the user-network. Secondly, one could investigate on topic models that would be updated continuously as user-generated-contents are captured. Indeed, topical evolutions highlight interest-shifts that might be leveraged for the activity prediction. Thirdly, one might investigate potential correlation between the activity and: (a) the users popularity levels; (b) the user communities, and (c) the topic evolutions. Finally, the unified representation for social-media might be used to tackle other *social-media-mining* problems within different social-media.

Bibliography

[1] Tom's hardware: Hardware news, tests and reviews. `www.tomshardware.com`. Accessed: 2014-09-01.

[2] Uci machine learning repository: Buzz in social media data set. `https://archive.ics.uci.edu/ml/datasets/Buzz+in+social+media+`. Accessed: 2014-09-01.

[3] *Proceedings of the Fifth International Conference on Weblogs and Social Media, ICWSM 2011, Barcelona, Catalonia, Spain, July 17-21, 2011*. The AAAI Press, 2011.

[4] *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '12, Beijing, China, August 12-16, 2012*. ACM, 2012.

[5] *Proceedings of the 21st World Wide Web Conference 2012, WWW 2012, Lyon, France, April 16-20, 2012*. ACM, 2012.

[6] *Proceedings of the Fifth International Conference on Web Search and Web Data Mining, WSDM 2012, Seattle, WA, USA, February 8-12, 2012*. ACM, 2012.

[7] *Proceedings of the Seventh International Conference on Weblogs and Social Media, ICWSM 2013, Cambridge, Massachusetts, USA, July 8-11, 2013*. The AAAI Press, 2013.

[8] *Proceedings of the Sixth ACM International Conference on Web Search and Data Mining, WSDM 2013, Rome, Italy, February 4-8, 2013*. ACM, 2013.

[9] *Proceedings of the Eighth International Conference on Weblogs and Social Media, ICWSM 2014, Ann Arbor, Michigan, USA, June 1-4, 2014*. The AAAI Press, 2014.

[10] Charu Aggarwal. *Managing and mining graph data*. Springer, New York, 2010.

[11] Charu Aggarwal. *Social network data analytics*. Springer, New York, 2011.

[12] Yong-Yeol Ahn, James P Bagrow, and Sune Lehmann. Link communities reveal multiscale complexity in networks. *Nature*, 466(7307):761–764, 2010.

[13] Yong-Yeol Ahn, Seungyeop Han, Haewoon Kwak, Sue B. Moon, and Hawoong Jeong. Analysis of topological characteristics of huge online social networking services. In *Proceedings of the 16th International Conference on World Wide Web, WWW 2007, Banff, Alberta, Canada, May 8-12, 2007*, pages 835–844. ACM, 2007.

[14] Sitaram Asur and Bernardo A. Huberman. Predicting the future with social media. In *International Conference on Web Intelligence, (WI 2010), Toronto, Canada, August 31 - September 3, 2010, Main Conference Proceedings*, pages 492–499. IEEE, 2010.

[15] Sitaram Asur, Bernardo A. Huberman, Gábor Szabó, and Chunyan Wang. Trends in social media: Persistence and decay. In *ICWSM 2011* [3].

[16] Thomas Aynaud, Eric Fleury, Jean-Loup Guillaume, and Qinna Wang. Communities in evolving networks: Definitions, detection, and analysis techniques. In *Dynamics On and Of Complex Networks, Volume 2*, pages 159–200. Springer, 2013.

[17] James P Bagrow. Evaluating local community methods in networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(05):P05001, 2008.

[18] D. Barber. *Bayesian Reasoning and Machine Learning*. Cambridge University Press, 2012.

[19] Evgeniy Bart, Rui Zhang, and Muzammil Hussain. Where would you go this weekend? time-dependent prediction of user activity using social network data. In *ICWSM 2013* [7].

[20] M. Basseville and I.V. Nikiforov. *Detection of abrupt changes: theory and application*. Prentice Hall Englewood Cliffs, NJ, 1993.

[21] M. Basseville, I.V. Nikiforov, et al. *Detection of abrupt changes: theory and application*, volume 104. Prentice Hall Englewood Cliffs, NJ, 1993.

[22] Fabrıcio Benevenuto, Gabriel Magno, Tiago Rodrigues, and Virgılio Almeida. Detecting spammers on twitter. In *7th Collaboration, Electronic messaging, Anti-Abuse and Spam Conference (CAES)*, volume 6, page 12, 2010.

[23] P. Berkhin. A survey of clustering data mining techniques. In *Grouping Multidimensional Data*, pages 25–71. Springer Berlin Heidelberg, 2006.

[24] Christopher M Bishop et al. *Pattern recognition and machine learning*, volume 1. springer New York, 2006.

[25] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

[26] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10):10008, 2008.

[27] John Adrian Bondy and Uppaluri Siva Ramachandra Murty. *Graph theory with applications*, volume 6. Macmillan London, 1976.

[28] Simon Bourigault, Cédric Lagnier, Sylvain Lamprier, Ludovic Denoyer, and Patrick Gallinari. Learning social network embeddings for predicting information diffusion. In *Seventh ACM International Conference on Web Search and Data Mining, WSDM 2014, New York, NY, USA, February 24-28, 2014*, pages 393–402. ACM, 2014.

[29] Danah m. Boyd and Nicole B. Ellison. Social network sites: Definition, history, and scholarship. *Journal of Computer-Mediated Communication*, 13(1):210–230, 2007.

[30] Danah Michele Boyd. *Taken out of context: American teen sociality in networked publics*. ProQuest, 2008.

[31] Fred Brauer and Carlos Castillo-Chavez. *Mathematical models in population biology and epidemiology*. Springer, 2011.

[32] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[33] Andrei Z. Broder, Ravi Kumar, Farzin Maghoul, Prabhakar Raghavan, Sridhar Rajagopalan, Raymie Stata, Andrew Tomkins, and Janet L. Wiener. Graph structure in the web. *Computer Networks*, 33(1-6):309–320, 2000.

[34] Chris Burges. From ranknet to lambdarank to lambdamart : An overview. Technical Report MSR-TR-2010-82, Microsoft Research, 2010.

[35] Olivier Chapelle, Yi Chang, and Tie-Yan Liu. Future directions in learning to rank. *Journal of Machine Learning Research*, 14:91–100, 2011.

[36] Jiyang Chen, Osmar R. Zaïane, and Randy Goebel. Local community identification in social networks. In *International Conference on Advances in Social Network Analysis and Mining, ASONAM 2009, 20-22 July 2009, Athens, Greece*, pages 237–242. IEEE Computer Society, 2009.

[37] Ahlame Douzal Chouakria and Panduranga Naidu Nagabhushan. Adaptive dissimilarity index for measuring time series proximity. *Advances Data Analysis and Classification (ADAC)*, 1(1):5–21, 2007.

[38] Aaron Clauset. Finding local community structure in networks. *Physical review E*, 72:026132, Aug 2005.

[39] Aaron Clauset, Cosma Rohilla Shalizi, and Mark EJ Newman. Power-law distributions in empirical data. *Society for Industrial and Applied Mathematics Review (SIREV)*, 51(4):661–703, 2009.

[40] Raviv Cohen and Derek Ruths. Classifying political orientation on twitter: It's not easy! In *ICWSM 2013* [7].

[41] William W. Cohen, Robert E. Schapire, and Yoram Singer. Learning to order things. *Journal Artificial Intelligence Research (JAIR)*, 10:243–270, 1999.

[42] Diane J Cook and Lawrence B Holder. *Mining graph data*. John Wiley & Sons, 2006.

[43] Michele Coscia, Giulio Rossetti, Fosca Giannotti, and Dino Pedreschi. Demon: a local-first discovery method for overlapping communities. In *KDD* [4], pages 615–623.

[44] Riley Crane and Didier Sornette. Robust dynamic classes revealed by measuring the response function of a social system. *Proceedings of the National Academy of Sciences*, 105(41):15649–15653, 2008.

[45] Aron Culotta. Towards detecting influenza epidemics by analyzing twitter messages. In *SOMA '10: Proceedings of the First Workshop on Social Media Analytics*, pages 115–122, New York, NY, USA, 2010. ACM.

[46] Maximilien Danisch, Jean-Loup Guillaume, and Bénédicte Le Grand. Multi-ego-centered communities in practice. *Social Network Analysis and Mining*, 2014.

[47] Easley David and Kleinberg Jon. *Networks, Crowds, and Markets: Reasoning About a Highly Connected World*. Cambridge University Press, 2010.

[48] Yajuan Duan, Long Jiang, Tao Qin, Ming Zhou, and Heung-Yeung Shum. An empirical study on learning to rank of tweets. In *COLING 2010, 23rd International Conference on Computational Linguistics, Proceedings of the Conference, 23-27 August 2010, Beijing, China*, pages 295–303. Tsinghua University Press, 2010.

[49] Maeve Duggan and Aaron Smith. *Social Media Update 2013*. Pew Research Center, 2013.

[50] Murat Dundar, Balaji Krishnapuram, Jinbo Bi, and R. Bharat Rao. Learning classifiers when the training data is not iid. In Manuela M. Veloso, editor, *IJCAI*, pages 756–761, 2007.

[51] Carl Eckart and Gale Young. The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3):211–218, 1936.

[52] Nicole B Ellison and Danah Boyd. Sociality through social network sites. *The Oxford handbook of internet studies*, page 151, 2013.

[53] Brian Everitt. *The Cambridge Dictionary of Statistics*. Cambridge University Press, Cambridge, UK New York, 2002.

[54] Michalis Faloutsos, Petros Faloutsos, and Christos Faloutsos. On power-law relationships of the internet topology. In *Proceedings of the ACM SIGCOMM 1999 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication, August 30 - September 3, 1999, Cambridge, Massachusetts, USA*, pages 251–262. ACM, 1999.

[55] Flavio Figueiredo. On the prediction of popularity of trends and hits for user generated videos. In *WSDM* [8], pages 741–746.

[56] Evelyn Fix and Joseph L Hodges Jr. Discriminatory analysis-nonparametric discrimination: consistency properties. Technical report, DTIC Document, 1951.

[57] Santo Fortunato. Community detection in graphs. *Physics Reports*, 486(3):75–174, 2010.

[58] Yoav Freund, Raj D. Iyer, Robert E. Schapire, and Yoram Singer. An efficient boosting algorithm for combining preferences. *Journal of Machine Learning Research*, 4:933–969, 2003.

[59] George W. Furnas, Scott C. Deerwester, Susan T. Dumais, Thomas K. Landauer, Richard A. Harshman, Lynn A. Streeter, and Karen E. Lochbaum. Information retrieval using a singular value decomposition model of latent semantic structure. In *Proceedings of the 11th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR'88, Grenoble, France, June 13-15, 1988*, pages 465–480. ACM, 1988.

[60] Maksym Gabielkov and Arnaud Legout. The complete picture of the twitter social graph. In *Proceedings of the 2012 ACM Conference on CoNEXT Student Workshop*, CoNEXT Student '12, pages 19–20, New York, NY, USA, 2012. ACM.

[61] Alexander Gammerman, Volodya Vovk, and Vladimir Vapnik. Learning by transduction. *The Computing Research Repository (CoRR)*, abs/1301.7375, 2013.

[62] Alberto Garcia. *Probability, statistics, and random processes for electrical engineering.* Pearson/Prentice Hall, Upper Saddle River, NJ, 2008.

[63] Stuart Geman, Elie Bienenstock, and René Doursat. Neural networks and the bias/variance dilemma. *Neural computation*, 4(1):1–58, 1992.

[64] Michelle Girvan and Mark EJ Newman. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences*, 99(12):7821–7826, 2002.

[65] Minas Gjoka, Maciej Kurant, Carter T. Butts, and Athina Markopoulou. A walk in facebook: Uniform sampling of users in online social networks. *The Computing Research Repository (CoRR)*, abs/0906.0060, 2009.

[66] Minas Gjoka, Maciej Kurant, Carter T Butts, and Athina Markopoulou. Practical recommendations on crawling online social networks. *Selected Areas in Communications, IEEE Journal on*, 29(9):1872–1892, 2011.

[67] Anna Goldenberg, Alice X Zheng, Stephen E Fienberg, and Edoardo M Airoldi. A survey of statistical network models. *Foundations and Trends in Machine Learning*, 2(2):129–233, 2010.

[68] Jacob Goldenberg, Barak Libai, and Eitan Muller. Talk of the network: A complex systems look at the underlying process of word-of-mouth. *Marketing letters*, 12(3):211–223, 2001.

[69] G.H. Golub and C.F. Van Loan. *Matrix Computations*. Johns Hopkins Studies in the Mathematical Sciences. Johns Hopkins University Press, 2013.

[70] Manuel Gomez-Rodriguez, David Balduzzi, and Bernhard Schölkopf. Uncovering the temporal dynamics of diffusion networks. In *Proceedings of the 28th International Conference on Machine Learning, ICML 2011, Bellevue, Washington, USA, June 28 - July 2, 2011*, pages 561–568. Omnipress, 2011.

[71] Mark Granovetter. Threshold models of collective behavior. *American journal of sociology*, pages 1420–1443, 1978.

[72] Mark S Granovetter. The strength of weak ties. *American journal of sociology*, pages 1360–1380, 1973.

[73] Jean-Loup Guillaume and Matthieu Latapy. Complex network metrology. *Complex systems*, 16(1):83, 2005.

[74] Adrien Guille and Hakim Hacid. A predictive model for the temporal dynamics of information diffusion in online social networks. In *Proceedings of the 21st World Wide Web Conference, WWW 2012, Lyon, France, April 16-20, 2012 (Companion Volume)*, pages 1145–1152. ACM, 2012.

[75] Adrien Guille, Hakim Hacid, Cécile Favre, and Djamel A. Zighed. Information diffusion in online social networks: a survey. *SIGMOD Record*, 42(2):17–28, 2013.

[76] Bo Han and Timothy Baldwin. Lexical normalisation of short text messages: Makn sens a #twitter. In *The 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference, 19-24 June, 2011, Portland, Oregon, USA*, pages 368–378. ACL, 2011.

[77] Trevor Hastie. *The elements of statistical learning data mining, inference, and prediction*. Springer, New York, 2009.

[78] Myles Hollander and Douglas A. Wolfe. *Nonparametric Statistical Methods, 2nd Edition*. Wiley-Interscience, 2 edition, 1999.

[79] Liangjie Hong, Amr Ahmed, Siva Gurumurthy, Alexander J. Smola, and Kostas Tsioutsiouliklis. Discovering geographical topics in the twitter stream. In *WWW 2012* [5], pages 769–778.

[80] Twitter Inc. About Twitter, Inc. `https://about.twitter.com/company`, 2014. [Online; accessed 1-May-2014].

[81] Anil K Jain, M Narasimha Murty, and Patrick J Flynn. Data clustering: a review. *ACM computing surveys (CSUR)*, 31(3):264–323, 1999.

[82] Thorsten Joachims. Optimizing search engines using clickthrough data. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, July 23-26, 2002, Edmonton, Alberta, Canada*, pages 133–142. ACM, 2002.

[83] Dan Jurafsky. *Speech and language processing*. Prentice Hall, Pearson Education International, Upper Saddle River, NJ u.a, 2014.

[84] Sanjay Ram Kairam, Dan J. Wang, and Jure Leskovec. The life and death of online groups: predicting group growth and longevity. In *WSDM* [6], pages 673–682.

[85] Andreas M. Kaplan and Michael Haenlein. Users of the world, unite! the challenges and opportunities of social media. *Business Horizons*, 53(1):59 – 68, 2010.

[86] François Kawala, Ahlame Douzal-Chouakria, Eric Gaussier, and Eustache Dimert. Prédictions d'activité dans les réseaux sociaux en ligne. In *4ième conférence sur les modèles et l'analyse des réseaux : Approches mathématiques et informatiques*, page 16, Grenoble, France, October 2013.

[87] François Kawala, Éric Gaussier, Ahlame Douzal, and Eustache Diemert. Apprentissage d'ordonnancement et influence de l'ambiguïté pour la prédiction d'activité sur les réseaux sociaux. In *CORIA 2014 - Conférence en Recherche d'Infomations et Applications, Nancy, France, March 19-23, 2014.*, pages 337–351, 2014.

[88] David Kempe, Jon M. Kleinberg, and Éva Tardos. Influential nodes in a diffusion model for social networks. In *Automata, Languages and Programming, 32nd International Colloquium, ICALP 2005, Lisbon, Portugal, July 11-15, 2005, Proceedings*, volume 3580 of *LNCS*, pages 1127–1138. Springer, 2005.

[89] Maurice George Kendall. *Rank correlation methods*. Griffin, 1948.

[90] Donald E. Knuth. *The Art of Computer Programming, Volume I: Fundamental Algorithms, 2nd Edition.* Addison-Wesley, 1973.

[91] Shoubin Kong, Qiaozhu Mei, Ling Feng, Fei Ye, and Zhe Zhao. Predicting bursts and popularity of hashtags in real-time. In *The 37th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '14, Gold Coast , QLD, Australia - July 06 - 11, 2014*, pages 927–930. ACM, 2014.

[92] Shoubin Kong, Qiaozhu Mei, Ling Feng, Zhe Zhao, and Fei Ye. On the real-time prediction problems of bursting hashtags in twitter. *The Computing Research Repository (CoRR)*, abs/1401.2018, 2014.

[93] Ravi Kumar, Jasmine Novak, and Andrew Tomkins. Structure and evolution of online social networks. In *Proceedings of the Twelfth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Philadelphia, PA, USA, August 20-23, 2006*, pages 611–617. ACM, 2006.

[94] Haewoon Kwak, Changhyun Lee, Hosung Park, and Sue B. Moon. What is twitter, a social network or a news media? In *Proceedings of the 19th International Conference on World Wide Web, WWW 2010, Raleigh, North Carolina, USA, April 26-30, 2010*, pages 591–600. ACM, 2010.

[95] Cédric Lagnier, Ludovic Denoyer, Éric Gaussier, and Patrick Gallinari. Predicting information diffusion in social networks using content and user's profiles. In *Advances in Information Retrieval - 35th European Conference on IR Research, ECIR 2013, Moscow, Russia, March 24-27, 2013.*, volume 7814 of *Lecture Notes in Computer Science*, pages 74–85. Springer, 2013.

[96] Cédric Lagnier, Éric Gaussier, and François Kawala. Modéliser l'utilisateur pour la diffusion de l'information dans les réseaux sociaux. *Ingénierie des Systèmes d'Information*, 17(3):1–22, 2012. Numéro spécial sur "Systèmes d'information : impact des réseaux sociaux".

[97] Himabindu Lakkaraju, Julian J. McAuley, and Jure Leskovec. What's in a name? understanding the interplay between titles, content, and communities in social media. In *ICWSM 2013* [7].

[98] Andrea Lancichinetti, Santo Fortunato, and János Kertész. Detecting the overlapping and hierarchical community structure in complex networks. *New Journal of Physics*, 11(3):033015, 2009.

[99] Andrea Lancichinetti, Filippo Radicchi, José J Ramasco, and Santo Fortunato. Finding statistically significant communities in networks. *PloS one*, 6(4):P18961, 2011.

[100] Daniel D Lee and H Sebastian Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, 1999.

[101] Sang Hoon Lee, Pan-Jun Kim, and Hawoong Jeong. Statistical properties of sampled networks. *Physical Review E*, 73(1):016102, 2006.

[102] Janette Lehmann, Bruno Gonçalves, Jose J. Ramasco, and Ciro Cattuto. Dynamical classes of collective attention in twitter. In *WWW* [5], pages 251–260.

[103] Jure Leskovec, Lars Backstrom, and Jon M. Kleinberg. Meme-tracking and the dynamics of the news cycle. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Paris, France, June 28 - July 1, 2009*, pages 497–506. ACM, 2009.

[104] Jure Leskovec, Jon M. Kleinberg, and Christos Faloutsos. Graph evolution: Densification and shrinking diameters. *Transactions on Knowledge Discovery from Data (TKDD)*, 1(1), 2007.

[105] Tie-Yan Liu. Learning to rank for information retrieval. *Foundations and Trends in Information Retrieval*, 3(3):225–331, 2009.

[106] Gilad Lotan, Erhardt Graeff, Mike Ananny, Devin Gaffney, Ian Pearce, and danah boyd. The arab spring — the revolutions were tweeted: Information flows during the 2011 tunisian and egyptian revolutions. *International Journal of Communication*, 5(0), 2011.

[107] Zhunchen Luo, Miles Osborne, Jintao Tang, and Ting Wang. Who will retweet me?: finding retweeters in twitter. In *The 36th International ACM SIGIR conference on research and development in Information Retrieval, SIGIR '13, Dublin, Ireland - July 28 - August 01, 2013*, pages 869–872. ACM, 2013.

[108] Zongyang Ma, Aixin Sun, and Gao Cong. Will this #hashtag be popular tomorrow? In *The 35th International ACM SIGIR conference on research and development in Information Retrieval, SIGIR '12, Portland, OR, USA, August 12-16, 2012*, pages 1173–1174. ACM, 2012.

[109] Fragkiskos D. Malliaros and Michalis Vazirgiannis. Clustering and community detection in directed networks: A survey. *Physics Reports*, 533(4):95 – 142, 2013. Clustering and Community Detection in Directed Networks: A Survey.

[110] C.D. Manning and H. Schütze. *Foundations of Statistical Natural Language Processing*. MIT Press, 1999.

[111] David A. McAllester, Tamir Hazan, and Joseph Keshet. Direct loss minimization for structured prediction. In *Advances in Neural Information Processing Systems 23: 24th Annual Conference on Neural Information Processing Systems, 6-9 December 2010, Vancouver, British Columbia, Canada*, pages 1594–1602. Curran Associates, Inc., 2010.

[112] Stanley Milgram. The small world problem. *Psychology today*, 2(1):60–67, 1967.

[113] Thomas P. Minka and John D. Lafferty. Expectation-propogation for the generative aspect model. *The Computing Research Repository (CoRR)*, abs/1301.0588, 2013.

[114] Alan Mislove, Massimiliano Marcon, P. Krishna Gummadi, Peter Druschel, and Bobby Bhattacharjee. Measurement and analysis of online social networks. In *Proceedings of the 7th ACM SIGCOMM Conference on Internet Measurement 2007, San Diego, California, USA, October 24-26, 2007*, pages 29–42. ACM, 2007.

[115] Mehryar Mohri. *Foundations of machine learning*. MIT Press, Cambridge, MA, 2012.

[116] Mohamed Morchid, Georges Linares, and Richard Dufour. Characterizing and predicting bursty events: The buzz case study on twitter. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation, LREC-2014, Reykjavik, Iceland, May 26-31, 2014*, pages 2766–2771. ELRA, 2014.

[117] Kevin P Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012.

[118] Mor Naaman, Hila Becker, and Luis Gravano. Hip and trendy: Characterizing emerging trends on twitter. *Journal of the American Society for Information Science and Technology*, 62(5):902–918, 2011.

[119] Nasir Naveed, Thomas Gottron, Jérôme Kunegis, and Arifah Che Alhadi. Bad news travel fast: a content-based analysis of interestingness on twitter. In *WebSci*, page 8. ACM, 2011.

[120] David F Nettleton. Data mining of social networks represented as graphs. *Computer Science Review*, 7:1–34, 2013.

[121] Blaise Ngonmang, Maurice Tchuente, and Emmanuel Viennet. Local community identification in social networks. *Parallel Processing Letters*, 22(1), 2012.

[122] Miles Osborne and Mark Dredze. Facebook, twitter and google plus for breaking news: Is there a winner? In *ICWSM 2014* [9].

[123] E. S. Page. Continuous inspection schemes. *Biometrika*, 41(1/2):100–115, 1954.

[124] Gergely Palla, Imre Derényi, Illés Farkas, and Tamás Vicsek. Uncovering the overlapping community structure of complex networks in nature and society. *Nature*, 435(7043):814–818, 2005.

[125] Gabriele Paolacci, Jesse Chandler, and Panagiotis G Ipeirotis. Running experiments on amazon mechanical turk. *Judgment and Decision making*, 5(5):411–419, 2010.

[126] Marco Pennacchiotti and Ana-Maria Popescu. A machine learning approach to twitter user classification. In *ICWSM 2011* [3].

[127] Sasa Petrovic, Miles Osborne, and Victor Lavrenko. Rt to win! predicting message propagation in twitter. In *ICWSM 2011* [3].

[128] Sasa Petrovic, Miles Osborne, Richard McCreadie, Craig Macdonald, Iadh Ounis, and Luke Shrimpton. Can twitter replace newswire for breaking news? In *ICWSM 2013* [7].

[129] Henrique Pinto, Jussara M. Almeida, and Marcos André Gonçalves. Using early view patterns to predict the popularity of youtube videos. In *WSDM* [8], pages 365–374.

[130] Ian Porteous, David Newman, Alexander T. Ihler, Arthur U. Asuncion, Padhraic Smyth, and Max Welling. Fast collapsed gibbs sampling for latent dirichlet allocation. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Las Vegas, Nevada, USA, August 24-27, 2008*, pages 569–577. ACM, 2008.

[131] Arnau Prat-Pérez, David Dominguez-Sal, Josep M. Brunat, and Josep-Lluis Larriba-Pey. Shaping communities out of triangles. In *21st ACM International Conference on Information and Knowledge Management, CIKM'12, Maui, HI, USA, October 29 - November 02, 2012*, pages 1677–1681. ACM, 2012.

[132] Arnau Prat-Pérez, David Dominguez-Sal, and Josep-Lluis Larriba-Pey. High quality, scalable and parallel community detection for large real graphs. In *WWW*, pages 225–236. ACM, 2014.

[133] Quantcast. Top Ranking International Websites. `https://www.quantcast.com/top-sites`, 2014. [Online; accessed 1-June-2014].

[134] Usha Nandini Raghavan, Réka Albert, and Soundar Kumara. Near linear time algorithm to detect community structures in large-scale networks. *Physical Review E*, 76(3):036106, 2007.

[135] Daniel Ramage, Susan T. Dumais, and Daniel J. Liebling. Characterizing microblogs with topic models. In *Proceedings of the Fourth International Conference on Weblogs and Social Media, ICWSM 2010, Washington, DC, USA, May 23-26, 2010*, 2010.

[136] Daniel M. Romero, Brendan Meeder, and Jon M. Kleinberg. Differences in the mechanics of information diffusion across topics: idioms, political hashtags, and complex contagion on twitter. In *WWW*, pages 695–704. ACM, 2011.

[137] Daniel M. Romero, Chenhao Tan, and Johan Ugander. On the interplay between social and topical structure. In *ICWSM 2013* [7].

[138] Guido Rossum. Python reference manual. Technical report, Amsterdam, The Netherlands, The Netherlands, 1995.

[139] Martin Rosvall and Carl T Bergstrom. Maps of random walks on complex networks reveal community structure. *Proceedings of the National Academy of Sciences*, 105(4):1118–1123, 2008.

[140] Matthew A Russell. *Mining the Social Web: Data Mining Facebook, Twitter, LinkedIn, Google+, GitHub, and More*. O'Reilly Media, Inc., 2013.

[141] Kazumi Saito, Masahiro Kimura, Kouzou Ohara, and Hiroshi Motoda. Learning continuous-time information diffusion model for social behavioral data analysis. In *Advances in Machine Learning*, volume 5828 of *Lecture Notes in Computer Science*, pages 322–337. Springer Berlin Heidelberg, 2009.

[142] Gerard Salton. *Automatic Text Processing: The Transformation, Analysis, and Retrieval of*. Addison-Wesley, 1989.

[143] Satu Elisa Schaeffer. Graph clustering. *Computer Science Review*, 1(1):27–64, 2007.

[144] Georgos Siganos, Sudhir Leslie Tauro, and Michalis Faloutsos. Jellyfish: A conceptual model for the as internet topology. *Journal of Communications and Networks*, 8(3):339–350, 2006.

[145] Bongwon Suh, Lichan Hong, Peter Pirolli, and Ed H. Chi. Want to be retweeted? large scale analytics on factors impacting retweet in twitter network. In *Proceedings of the 2010 IEEE Second International Conference on Social Computing, SocialCom Minneapolis, Minnesota, USA, August 20-22, 2010*, pages 177–184. IEEE Computer Society, 2010.

[146] Gábor Szabó and Bernardo A. Huberman. Predicting the popularity of online content. *Commun. ACM*, 53(8):80–88, 2010.

[147] Yla R Tausczik and James W Pennebaker. The psychological meaning of words: Liwc and computerized text analysis methods. *Journal of language and social psychology*, 29(1):24–54, 2010.

[148] Yee Whye Teh, Kenichi Kurihara, and Max Welling. Collapsed variational inference for HDP. In *Advances in Neural Information Processing Systems 20, Proceedings of the Twenty-First Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 3-6, 2007*, pages 1481–1488, 2007.

[149] Kurt Thomas, Chris Grier, Dawn Song, and Vern Paxson. Suspended accounts in retrospect: an analysis of twitter spam. In *Proceedings of the 11th ACM SIGCOMM Internet Measurement Conference, IMC '11, Berlin, Germany, November 2-, 2011*, pages 243–258. ACM, 2011.

[150] Richard Trudeau. *Introduction to graph theory*. Dover Publications, New York, 1993.

[151] Oren Tsur and Ari Rappoport. What's in a hashtag?: content based prediction of the spread of ideas in microblogging communities. In *WSDM 2012* [6], pages 643–652.

[152] Ibrahim Uysal and W. Bruce Croft. User oriented tweet ranking: a filtering approach to microblogs. In *Proceedings of the 20th ACM Conference on Information and Knowledge Management, CIKM 2011, Glasgow, United Kingdom, October 24-28, 2011*, pages 2261–2264. ACM, 2011.

[153] Hamed Valizadegan, Rong Jin, Ruofei Zhang, and Jianchang Mao. Learning to rank by optimizing ndcg measure. In *Advances in Neural Information Processing Systems 22: 23rd Annual Conference on Neural Information Processing Systems 2009. Vancouver, British Columbia, Canada, 7-10 December 2009,*, pages 1883–1891. Curran Associates, Inc., 2009.

[154] Graham Vickery and Sacha Wunsch-Vincent. *Participative web and user-created content: Web 2.0 wikis and social networking*. Organization for Economic Cooperation and Development (OECD), 2007.

[155] Fei Wang, Tao Li, Xin Wang, Shenghuo Zhu, and Chris H. Q. Ding. Community discovery using nonnegative matrix factorization. *Data Mining and Knowledge Discovery*, 22(3):493–521, 2011.

[156] Stanley Wasserman and Katherine Faust. *Social network analysis: methods and applications*. Cambridge University Press, 1994.

[157] Lilian Weng, Filippo Menczer, and Yong-Yeol Ahn. Predicting successful memes using network and community structure. In *ICWSM 2014* [9].

[158] Douglas Brent West. *Introduction to graph theory*, volume 2. Prentice hall Upper Saddle River, 2001.

[159] Ian H Witten and Eibe Frank. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2005.

[160] Guang Xiang, Zeyu Zheng, Miaomiao Wen, Jason I. Hong, Carolyn Penstein Rosé, and Chao Liu. A supervised approach to predict company acquisition with factual

and topic features using profiles and news articles on techcrunch. In *Proceedings of the Sixth International Conference on Weblogs and Social Media, Dublin, Ireland, June 4-7, 2012*. The AAAI Press, 2012.

[161] Jierui Xie, Stephen Kelley, and Boleslaw K Szymanski. Overlapping community detection in networks: The state-of-the-art and comparative study. *ACM Computing Surveys (CSUR)*, 45(4):43, 2013.

[162] Jaewon Yang and Jure Leskovec. Modeling information diffusion in implicit networks. In *ICDM 2010, The 10th IEEE International Conference on Data Mining, Sydney, Australia, 14-17 December 2010*, pages 599–608. IEEE Computer Society, 2010.

[163] Jaewon Yang and Jure Leskovec. Patterns of temporal variation in online media. In *Proceedings of the Forth International Conference on Web Search and Web Data Mining, WSDM 2011, Hong Kong, China, February 9-12, 2011*, pages 177–186. ACM, 2011.

[164] Jaewon Yang and Jure Leskovec. Overlapping community detection at scale: a nonnegative matrix factorization approach. In *WSDM* [8], pages 587–596.

[165] Lei Yang, Tao Sun, Ming Zhang, and Qiaozhu Mei. We know what @you #tag: does the dual role affect hashtag adoption? In *WWW 2012* [5], pages 261–270.

[166] G Yi, S Coleman, and Q Ren. Cusum method in predicting regime shifts and its performance in different stock markets allowing for transaction fees. *Journal of Applied Statistics*, 33(7):647–661, 2006.

[167] Reza Zafarani. *Social media mining : an introduction*. Cambridge University Press, New York, 2014.

[168] Peng Zhang, Xufei Wang, and Baoxin Li. On predicting twitter trend: factors and models. In *Advances in Social Networks Analysis and Mining 2013, ASONAM '13, Niagara, ON, Canada - August 25 - 29, 2013*, pages 1427–1429. ACM, 2013.

[169] Yongzheng Zhang and Marco Pennacchiotti. Predicting purchase behaviors from social media. In *22nd International World Wide Web Conference, WWW '13, Rio de Janeiro, Brazil, May 13-17, 2013*, pages 1521–1532, 2013.

[170] Yu Zhang and Dit-Yan Yeung. Overlapping community detection via bounded nonnegative matrix tri-factorization. In *KDD 2012* [4], pages 606–614.

[171] Wenjun Zhou, Hongxia Jin, and Yan Liu. Community discovery and profiling with social messages. In *KDD 2012* [4], pages 388–396.

[172] Zhi-Hua Zhou. *Ensemble methods : foundations and algorithms*. Taylor & Francis, Boca Raton, FL, 2012.