Recommender Systems in the Era of Sentiment Analysis and Social Media

Davide Feltoni Gurini

Roma Tre University

A thesis submitted for the degree of

Doctor of Philosophy

Dec 2016

Acknowledgements

This thesis is dedicated to my incredible wife and to my beloved grandmother.
You will always be a source of inspiration and motivation for my life.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

With the explosion of Web applications such as blogs, forums, and social networking sites, the users' online activities have been changed. The new generation of Web contents are no longer read-only, but Web users are nowadays the real "producers of information". They actively participate in social networks, upload their personal photos, create information, and establish new relationships with hundreds of new virtual friends. The millions of online users spend hours daily in these sites, and generate rich information and various new sources of knowledge that has not been available before. Currently, the most popular online social networks are *Twitter*[1], *Facebook*[2], *LinkedIn*[3], and *Instagram*[4], while in September 2016 just Facebook counts more than 1,7 billion [5] users that generate 4 Peta byte of data and 345 billion *likes* per day. These numbers show the growing popularity of social networking sites that currently provide users with huge volumes of information, and hence pose new great challenges and research issues for developing accurate information retrieval methods, data search and mining, and recommender systems (RSs).

Recommender systems has became an autonomous research area in the second part of 1990s [8] and have attracted much attention from multiple field, such as mathematics, physics, psychology, and computer science [47]. Many techniques are considered for building RSs, which can be generally classified into content-based methods, collaborative filtering (CF) based methods, and hybrid methods.

RSs has evolved in the last decade, with the international research community that is currently focusing on how to exploit these established methodologies in the context of social networks tackling the information overload problem. A typical information overload

[1] `http://www.twitter.com`
[2] `http://www.facebook.com`
[3] `www.linkedin.com`
[4] `www.instagram.com`
[5] `https://www.statista.com/`

problem in social networks - might occur - when a online user needs to keep himself updated about a specific topic or on a certain expert in that topic. For example a user wants to remain updated about the topic technology. Searching Twitter using the keyword "technology" the research will return million of different content and users that wrote about technology. One of the main task of Recommender Systems, is to tackle this kind of information overload problem by filtering and suggesting information that is of potential interest to online users. RSs growth and social networks popularity also create new challenges and opportunities, with various benefits for both research areas. One of the major added value of social platforms is to encourage interaction between users, and this interaction can be extracted and used as an input to the RS, as it helps to better understand the users' interests and the information needs. Moreover, the structure of the fundamental network in a social platform can contribute to generate recommendations that are more trusted by users (e.g. by considering the number of common friends, as generally people trust more recommendations from closer friends). In fact online social relations provide a different way for individuals to communicate digitally and allow online users to share ideas and opinions with their connected users. A user's preference is similar to, or is influenced by their online friends, and this can be explained by social correlation theories such as homophily [124] and social influence [122]. Homophily indicates that users with similar preferences are more likely to be connected, and social influence reveals that connected users are more likely to have similar preferences. Analogous to the fact that people in real world are likely to seek suggestions from their friends before making a purchase decision and users' friends effectively provide good recommendations, social relations can be potentially exploited to improve the performance of online recommender systems. Furthermore, RSs can clearly help improve user participation in social systems, as they can recommend new friends or interesting content. Thus, the user will be more motivated to keep on-going participation in the social platform, because the more content a user shares, the more relevant connections the system can recommend, having a precise profile about him or her. Using this connection between social platforms and RSs, new scenarios can be defined for advanced applications, such as recommending items [78], tags [158], people [194, 81], news [117], topics [63], and communities [35]. Therefore, as resumed in Figure 1.1, we can certainly affirm that the social web provides a huge opportunity for improving RSs, and vice-versa.

In this scenario, with the expanding demand of RSs on social networks, detecting sentiments and opinions from the Web is becoming an increasingly widespread and important form of data interpretation. In particular, Sentiment Analysis (SA) or Opinion Mining aims to understand subjective information, such as opinions, points of views and feelings

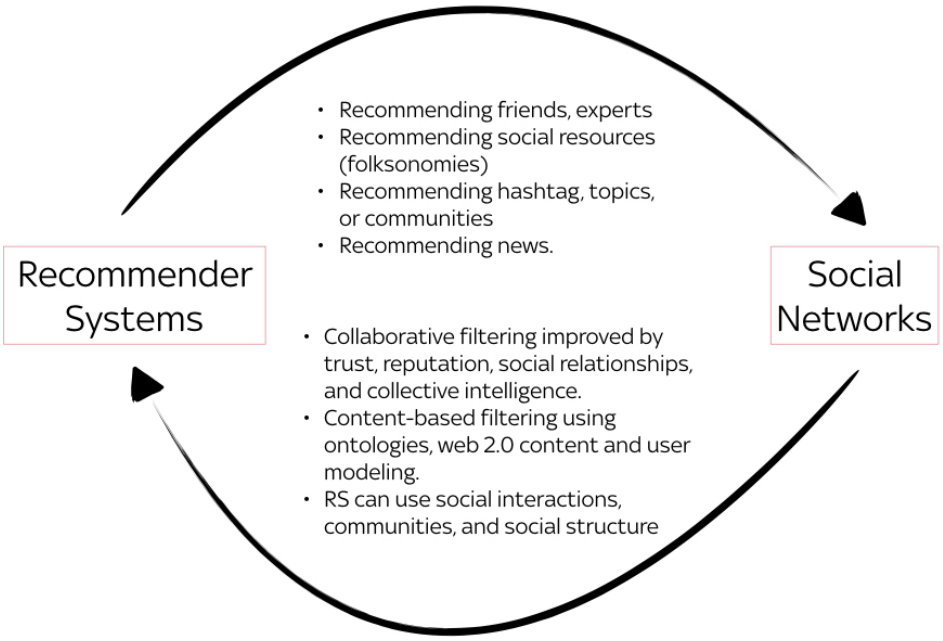


Figure 1.1: A resume of the mutual contribution between Recommender Systems and Social Media.

expressed by users in the content they generate. SA algorithms identify how positive, negative or neutral is the produced content regarding a specific entity, that is, a product, person, organization, event and topic [116]. This research area is a field at crossroad of Information Retrieval and Natural Language Processing (NLP) which has become extremely popular in the last years in terms of research attention, industry consideration, and online studies as showed in Figure 1.2, mainly thanks to the advent of microblogging platforms such as Twitter. As a matter of fact, Sentiment Analysis in the field of social media permits companies, marketers, organizations, or individuals, to understand their business online reputation, identify public opinions regarding products and services of themselves or their competitors, and gain insights about possible emerging trends and changes in market opinions, or identify crisis.

The main rationale behind this thesis is that Sentiment Analysis can improve the performance of RS in social media. Although a large number of contributions have been devoted to the people-to-people recommendation issue, by exploiting Sentiment Analysis of user-generated contents for the purposes of user recommendation task has not been deeply investigated yet. Xu *et al.* [183] transform the sentiment community discovery into a corre-

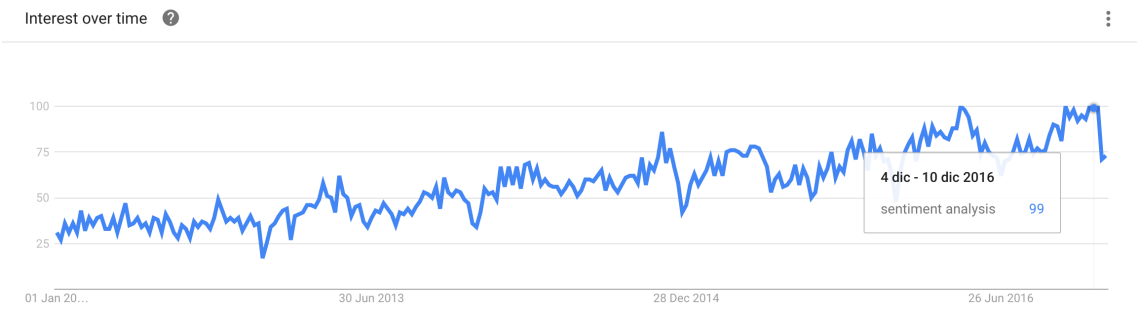


Figure 1.2: The Google search evolution trend of Sentiment Analysis topic.

lation clustering problem and propose a random rounding algorithm based on semidefinite programming for its solution. In [132] the authors describe an unsupervised approach based a non-parametric clustering algorithm for detecting hyper-groups of communities, called *hyper-communities*, where users share the same sentiments. Singh *et al.* [159] introduce a hybrid recommender system that improves the results of collaborative filtering by incorporating a sentiment classifier in the movie recommendation scenario. While research focused on the aforementioned topics, there are few attempts that consider SA for a user recommender. Therefore, in this thesis we pose several research questions, the main ones listed as follows:

- How Sentiment Analysis can be leveraged to build a RS and improve its precision?
- How considering temporal dynamics shape the performance of a sentiment-based RS?
- Are there differences depending on the category of topics dealt with by the user?

These research questions and others more specific points are discussed in this dissertation and explained in details afterwards in this chapter.

This thesis consists of eight chapters. After introducing the motivation underlying this research thesis in Chapter 1, in Chapter 2 are presented the general backgrounds and related works. The main contributions of this thesis are described in Chapter 3-7, each of which start with a motivation of the research questions that are investigated in the corresponding chapter and conclude with a summary of main findings and contributions. The Chapter 8 provides a summary of the research contributions and outcomes discovered in this dissertation. In summary, this thesis contributes to research in the areas described below.

Chapter 3: Sentiment-based User Recommender on Twitter. In this chapter is proposed a new weighting function that takes into consideration Sentiment Analysis of posts, with the aim to improve the recommendation task. The rationale behind this section is that users in social networks may share similar interests but might have different opinions on them. As a result, considering the contribution of user sentiments can yield benefits in recommending possible friends to follow. In this section firstly we devised a proprietary algorithm of sentiment analysis, that is specific for Twitter analysis. Secondly, we propose a user recommendation technique based on a different weighting function, we named *sentiment-volume-objectivity (SVO)* function, which takes into account not only user interests, but also sentiments toward them. Such function allows us to build richer user profiles to employ in the recommendation process than other content-based approaches. The main research question we advance in this chapter is: can the consideration of this novel sentiment-based function yield benefits to the user RS?

Chapter 4: Exploiting Signals and Temporal Dynamics for a People-to-People RS. In this chapter is introduced a novel framework with a new user model, called *bag-of-signals*, that represents how user interests vary over time for creating more comprehensive user profiles. The basic idea underlying such approach is to represent each user interest as a signal. In order to analyze such signals we make use of the wavelet transform, a signal processing technique that captures the frequency content of any signal, together with their precise location of occurrence in the time domain. After evaluating the performance of this techniques we consider another signal dimension that represents the sentiment of a user toward a specific topic. The Sentiment Analysis model is build as the previous Chapter 3. The research questions we pose are: (i) can the consideration of temporal patterns of changing users interests really impact the characteristics and quality of user recommender? (ii) Can Sentiment Analysis yield some benefits to the proposed temporal-based RS?

Chapter 5: Leveraging Community Detection Techniques for User RS. From the evaluation results of the previous sections obtained in Chapter 3, Sentiment Analysis has preliminary proved its benefits for a people RS. In this Chapter we want therefore to build a more complex user Recommender System that can exploit the potentials of SA and social networks, considering also how topic evolve and change in user comments. To reach this goal, we propose a new approach for realizing user recommenders, named *SCORES (Sentiment COmmunities REcommender System.)* This algorithm relies on the identification of sentiment communities in which, for each topic cited by the user, we consider not only the relative sentiment, but also the *SVO* of contents generated by him. The graph is built

by considering each topic discussed by the users as a vertex and the edges are generated by considering the Tanimoto similarity between users. Clustering based on the modularity optimization allows us to detect the latent communities. The recommendation process occurs by suggesting to the target user the most similar K users based on several tie strength measures. The research questions we set are: (i) which is the best graph techniques that enhance the contribution of the Sentiment Analysis? (ii) Can sentiment improve the final recommendation precision? (iii) Are there differences depending on the category of topics dealt with by the user?

Chapter 6: Matrix Factorization Recommender System. To address scalability issues and temporal dynamics we propose a novel recommendation engine, that still relies on the identification of semantic attitudes, that is, sentiment, volume, and objectivity extracted from user-generated content. In order to do this at large-scale on traditional social networks, we devise a three-dimensional matrix factorization, one for each attitude. Potential temporal alteration of users' attitudes are also taken into consideration in the factorization model. This chapter also represents one of the first attempt to combine sentiment in a matrix factorization recommender systems. Research questions that we want to answer are (i) does content published by users and, in particular, the inferred attitudes, allows for a better identification of potential relationships between users? How does temporal analysis of these attitudes impact the recommendation? Furthermore the scientific contributions coming from this section also include a comparative experimental results of a set of different evaluation metrics, including a range of non-accuracy measures, such as diversity and novelty and an extensive evaluation of the proposed algorithm on real world datasets.

Chapter 7: A Sentiment-based Youtube Video Recommender. To understand whether Sentiment Analysis enriches the recommendation process also in others social networks, we explore similar techniques for video recommendation on Youtube. Youtube social network is a specific video sharing network, where comments left by the viewers often provide valuable information to describe sentiments, opinions and tastes of the users. For this reason, we propose a novel re-ranking approach that takes into consideration that information in order to provide better recommendations of related videos. The research question we pose in this Chapter is therefore: how much Sentiment Analysis can enrich the recommendation process on Youtube? A preliminary evaluation highlights an increase of the recommender precision compared with a state-of-the-arts approach.

1.1 Origin of Chapters

Each of the main presented chapters (Chapter 3-7) is based on at least one peer-reviewed publication, which has been published in conferences or journals that are related to the research topics of this thesis.

Chapter 3 This chapter contains material from one paper published in *2013 RSWeb at ACM Recommender Systems Conference* [50]

Chapter 4 This chapter comprises and summarizes our findings, which are presented in two paper published in *SRS 2016 at the 22nd International Conference on World Wide Web* [15] and in *INRA 2016, 4th International Workshop on News Recommendation and Analytics in conjunction with UMAP 2016* [29]

Chapter 5 This chapter contains findings and research coming from a work published in three papers: *UMAP 2014, the 22nd International Conference on User Modeling, Adaptation, and Personalization* [71], *International Conference on Web Information System Engineering* [73], and *SPS 2015, International Workshop on Social Personalisation & Search in conjunction with SIGIR* [72]

Chapter 6 This chapter contains a work published in the 2017 Journal edition of *Future Generation Computer Systems* [74]

Chapter 7 This chapter comprises a work published in *RecSys 2015 Poster Proceedings* [57]

Chapter 2

Background

In this chapter are introduced the research area addressed in this thesis. Firstly, is introduced the Recommender System research field, with an overview of the techniques and methods that are used in this work. Secondly, is given a resume of Social Network domain, with a focus on the community detection and link prediction tasks. Finally, is presented an essay regarding Sentiment Analysis, providing definition and methodologies that are useful to the comprehension of the following chapters.

2.1 Recommender Systems

2.1.1 Overview

Recommender Systems (RSs) are software tools and techniques providing suggestions for items to be of use to a user [149]. The suggestions is regarding various decision-making processes, such as what items to buy, what hotel to choose, or what user connect with into social networks. The information can be acquired explicitly - typically by collecting users' ratings - or implicitly by monitoring users' behavior, such as songs heard, applications downloaded, users' social timeline, and web sites visited. RSs may use demographic features of users (like age, nationality, gender), psychological features [133], and social information, like followers, followed, tweets and posts, that are commonly used in Web 2.0. It is also growing the use of information from Internet of things (e.g., GPS locations, RFID [175], real-time health signals). As a matter of fact, RS implementation in the Internet has recently increased, which has facilitated its use in different areas. The most common research papers are focused on movie recommendation studies [148, 152], however a great volume of literature for RS is centered on different topics, such as music [31, 23], television [161], books [134], documents [144, 143], scientific documents [42], e-learning [61], e-commerce [91], applications in markets [104] and social media[3, 33, 15], among others.

Accordingly to the taxonomy provided by [149] there are six different classes of RS techniques: content-based, collaborative filtering, demographics, knowledge-based, community-based, and hybrid.

In the following sections will be discussed the two most used techniques such as content-based and collaborative filtering.

2.1.2 Content-Based

Content-based (CB), also referred to as cognitive filtering, recommends items based on a comparison between the content of the items and a user profile. The content of each item is represented as a set of descriptors or terms, typically the words that occur in a document. The user profile is represented with the same terms and built up by analyzing the content of items which have been seen by the user. The basic idea in CB recommendation is to recommend items that are similar to those items that the user has liked or that have expressed an interest in the past [20]. The recommendation process basically consists in matching up the attributes of the user profile against the attributes of a content object. The result is a relevance judgment that represents the user's level of interest in that object. If a profile accurately reflects user preferences, it is of tremendous advantage for the effectiveness of an information access process [118]. Items to be recommended can be very different depending on the number and types of attributes used to describe them. Each item can be described through the same small number of attributes with known set of values, such as Web pages, social network posts, news, emails or documents, described through unstructured text. In that case there are no attributes with well-defined values, and the use of document modeling techniques with origin in information retrieval research is necessary, furthermore this kind of textual features may create a number of issues when learning a user profile, mostly due to the ambiguity of the natural language. As the content-based approach has its roots in information filtering and information retrieval research [19], most CB systems use retrieval models such as the vector space model to construct item profiles as well as user profiles. Items or in some case users that can be recommended to the target user are represented by a set of features, also called attributes or properties. For example, in people recommender systems the item profile of user u, denoted as $UserProfile(u)$, can be represented by a vector in a multi-dimensional space, where each dimension corresponds to an interest that the user u has implicit or explicit expressed in the past. Various different weighting schemes such as TF or $TFxIDF$ can be applied to determine the weight of each element in the item profile and compute the similarity between users. The recommendation process consists of three steps [118]. The fist step focuses on analyzing the content of items to extract relevant structured information for the next steps. The main responsibility of the second step is to

construct item profiles, which exploit a set of properties to characterize items, as well as user profiles that describe users' tastes, preferences, and information needs. Finally, the RS tries to find relevant items for a user by matching the user profile against the profiles of items to be recommended.

The content-based method has several advantages described as follows [118]:

- **User independence.** In comparison to collaborative filtering techniques, CB recommender systems exploit solely the history of a user to construct the user profile for the computation of recommendations. Therefore, the algorithm does not require any extra information from other users.

- **Transparency.** Since both item and user profiles are constructed with features that are extracted from the content, a content-based recommender system allows for providing explanations on how the system works by describing explicitly the features that cause a particular recommendation. In contrast, the only explanation that can be provided for an recommendation based on collaborative filtering is that some (unknown) users with similar preferences liked that item [20]. The explicit explanations can help users judge whether they should trust the recommendations. For example, Cramer et al. conduct a user study to investigate the impact of transparency on user trust in content-based RSs [38]. They discover that providing explicit explanations to users increases their acceptance of the systems.

- **New Item.** In content-based RSs, it is possible to recommend items that are not yet rated by any user. Therefore, the systems do not suffer from the new item problem. The content-based techniques can be applied to recommend emerging items such as Twitter messages related to breaking news [142]. In contrast, in collaborative systems, new items need to be rated by a substantial number of users in order to generate accurate recommendations.

However, content-based RSs also suffer of some disadvantages that are explained as follows [8]:

- **Limited Feature.** This approaches are limited by the number and type of features that are used to represent the items to be recommended. Therefore, content-based systems need to first extract features from the content of items to construct item and user profiles. In many applications, the feature extraction requires domain knowledge or ontologies [127].

- **Content Similarity.** For each user, content-based approach tries to identify the most similar items based on the user profile to compute recommendations. This results in a lack of serendipity, i.e., the recommendations may have a limited degree of novelty. This problem was addressed by [8].

- **User Cold-start.** In order to understand users' preferences and deliver accurate recommendations, a content-based RSs needs to collect sufficient number of ratings for each user in the system. In different approaches that takes into account of implicit feedback such as the user timeline on social network, the recommender need to gather enough social post to infer the user profile. As a consequence, for a (new) user who only has few ratings or few social posts, the system is not capable of constructing the user profile and further providing reliable recommendations.

- **Semantic Challenge.** Textual features create a number of complications when learning a user profile, due to the natural language processing and its ambiguity. One of the possible problem is that traditional keyword-based profiles are unable to capture the semantics of user interests because they are primarily driven by a string matching operation. This approach can suffer problems of synonymy that can reduce the precision of the recommender.

The information source that content-based filtering systems are mostly used with are text documents. A standard approach for term parsing selects single words from documents. The vector space model and latent semantic indexing are two methods that use these terms to represent documents as vectors in a multi-dimensional space. Relevance feedback, genetic algorithms, neural networks, and the Bayesian classifier are among the learning techniques for learning a user profile. The vector space model and latent semantic indexing can both be used by these learning methods to represent documents. Some of the learning methods also represent the user profile as one or more vectors in the same multi-dimensional space which makes it easy to compare documents and profiles. Other learning methods such as the Bayesian classifier and neural networks do not use this space but represent the user profile in their own way.

2.1.3 Collaborative Filtering

The core idea of most Collaborative Filtering (CF) recommender systems is based on categorizing users with similar interests and then recommending items that similar users like. The system performs a comparison between the target user and other users based on their

ratings, and predicts the rates of unseen items in a target user's profile. Accordingly, it orders the list of items based on scores, and suggests items with the highest predicted ratings. This technique is called *collaborative filtering (CF)*, and it is widely applied in recommender systems processes such as in people recommendation on Twitter [76]. Given an unrated entry of an item by a user to be evaluated, CF tries to find other users similar to target user (user-based), or, other items similar to target item (item-based). Then, the unknown rating is predicted by weighting the known ratings of the target item by similar users, or, the known ratings of similar items by the target user. It is based on the assumption that similar users have the same interests and the target user will like the items which he/she has selected before [182]. Breese [27] introduces a classification of CF algorithms that divides them into two broad classes: memory- based algorithms and model-based algorithms.

Memory-based algorithms essentially are heuristics that make rating predictions based on entire collection of previously rated items by the users. That is, the value of the unknown rating $r_{u,i}$ for user u and item i is usually computed as an aggregate of the ratings of some other (usually, the N most similar) users for the same item i. In general, there are two ways to compute recommendations in memory-based systems: user-based and item-based filtering. *User-based filtering* first matches the target user profile of a user against the user profiles of other users in the system to identify a set of users (neighbors) who have similar preferences. Then the interest of that user for a new item is evaluated by aggregating the ratings given by the top-k most similar users for the same item [45]. In user-based systems, user profiles are usually represented as vectors. Then the similarities between user profiles can be measured using metrics such as cosine similarity or Pearson correlation coefficient [148]. These metrics can also be used to compute the similarities between items. *Item-based filtering* estimates the interest of a user for a new item based on the ratings of the most similar items in the system [44, 152].

Model-based algorithms [107, 99, 194, 185, 111, 119] use the collection of ratings to learn a model, which is then used to make rating predictions. Practically , model-based recommendation algorithms apply machine learning techniques to learn a predictive model based on a user-item matrix. The goal is to identify latent factors which are used to model the user-item interactions in a system. The model is trained using existing data and then applied to compute recommendations. Bresse et al. [27] investigate two probabilistic approaches for learning the model: clustering and Bayesian network. Another group of model-based algorithms, which becomes popular through the Netflix competition [107], is based on matrix factorization techniques (MF) such as Singular Value Decomposition (SVD) and Latent Dirichlet Allocation (LDA) [25]. In the next section will be a focus on

MF techniques since was adopted in this work. CF techniques has several known advantages as follows:

- **Simple and efficient.** Most of CF methods are intuitive and relatively simple to build. E.g. Neighborhood-based methods in the simplest form need the tuning of just one parameter (the number of neighbors used for prediction) [45].
- **Complexity Reductions.** CF techniques scales well increasing the number of items and users. As a matter of fact CF recommender is not increasing the complexity while increase the number of feature, that is one of the possible disadvantages of content-based systems when dealing with million of textual features.
- **Novelty and Serendipity.** CF recommender usually improves the novelty of the recommendation and the serendipity, that is, are capable to suggest novel items, even the ones that are not similar to those which have been rated in the past [8].

Nonetheless CF techniques have their own disadvantages, that are summarized as follows:

- **User Cold-start.** Collaborative filtering systems suffer from the new user problem, i.e., the systems would not be able to learn the preferences of a user and make accurate recommendations until the user gives a substantial number of ratings. Several recommendation systems employ hybrid approach, which combines collaborative filtering and content-based techniques, to address this problem [154, 28]. This problem is common also for content-based recommender and for the recommendation task in general.
- **New Item.** Since collaborative filtering methods rely on using other users' activities to estimate the interest of a given user for an item. Therefore, the item must have been rated or seen by other users in order to compute recommendations. Moreover, in many collaborative filtering systems, most users only interact with a very small fraction of all items, which makes the user-item matrices immensely sparse. Due to the lack of available information such as users' ratings the quality of recommendations may not be satisfying.

Hybrid RSs combine the advantages of the aforementioned CF and content-based techniques, with the aim to improve the recommendation system. [28] give also a formalization of seven different type of hybrid recommender: weighted, switching, mixed, feature combination, meta-level, cascade, and feature augmentation. From combining two algorithm

Category	Techniques	Main Advantages	Main Failure
MEMORY-BASED CF	– Neighbor-based CF (item-based/user-based CF algorithms with Pearson/vector cosine correlation) – Item-based or User-based top-N recommendation	– easy implementation – not consider content of the recommended item – scale well with co-rated items	– depends on human ratings – data sparsity causes decrease of performance – limited scalability for large datasets – user and item cold start problem
MODEL-BASED CF	– Bayesian CF – clustering CF – MDP based CF – latent semantic CF – sparse factor analysis – dimensionality reduction (e.g. SVD, MF)	– better address scalability and sparsity problems – usually improve prediction performance – intuitive rationale of recommendations	– expensive model building – trade-off between performance and scalability – may lose useful information with dimension reduction
HYBRID	– content-based CF, for example Fab[20] – content-boosted CF – hybrid combining content and model-based CF algorithms	– overcome CF limitation with single recommender – usually improve prediction performance – overcome CF problem such as sparsity and gray sheep	– increase complexity and expensive model building – need external further information that are not always available

Table 2.1: Overview of the collaborative filtering techniques

in sequence, or joining the feature of two approaches, or just switching the specific method depending on the case in order to improve the performances, hybrid methods are largely employed especially for Social Network recommendation tasks such as people recommendation [115, 2, 79]. Furthermore, Torres et al. [166] found that in some application an hybrid method may outperform individual algorithms.

In Table 2.1 are finally resumed some of the methodologies for CF, CB and hybrid recommender systems.

2.1.4 Matrix Factorization

Some of the most successful realizations of latent factor models are based on matrix factorization. In its basic form represented in Figure 2.1, matrix factorization characterizes both items and users by vectors of factors inferred from item rating patterns. These methods have become popular in recent years by combining good scalability with predictive accuracy, and in addition they offer much flexibility for modeling various real-life situations. Recommender systems rely on different types of input data, which are often placed in a matrix with one dimension representing users and the other dimension representing items of interest. Matrix factorization models map both users and items to a joint latent factor space of dimensionality f such that ratings are modeled as inner products in that space.

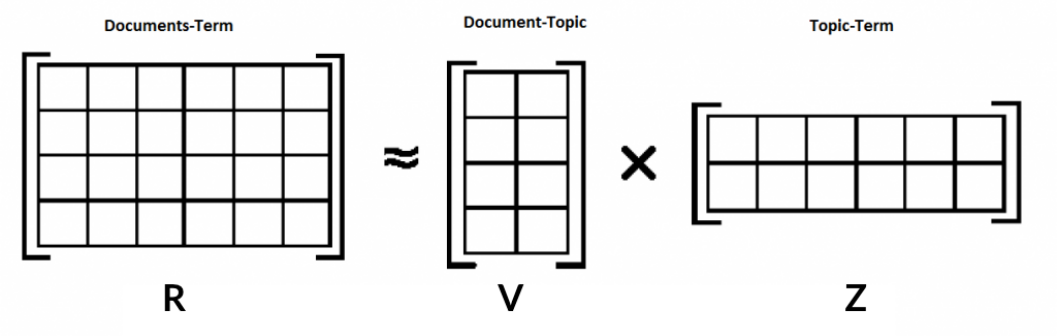


Figure 2.1: Basic form of matrix factorization. R is m by n rating matrix. U is user factor matrix and Z is item factor matrix.

Accordingly, each user u is associated with a vector $p_u \in R^f$ and each item i is associated with a vector $q_i \in R^f$. A rating is predicted by the rule [107]:

$$r_{u,i} = q_i^T p_u \tag{2.1}$$

In order to learn the vectors p_u and q_i, the system minimizes the regularized squared error on the set of known ratings. The constant λ controls the extent of regularization, as usually determined by cross validation.

$$min_{q^*,p^*} \sum_{(u,i)\in K} (r_{ui} - q_i^T p_u)^2 + \lambda(\left\|q_i\right\|^2 \left\|p_u\right\|^2) \tag{2.2}$$

The (u, i) pairs for which $r_{u,i}$ is known are stored in the set $K = (u, i)|r_{u,i}$ [106].

It is also possible to include a bias parameter into the factorization:

$$b_{ui} = \mu + b_u + b_i \tag{2.3}$$

where the parameter b_u and b_i are the observed deviation of user u and item i from the average values. To give an example referring to 2.4, we would like to predict an estimation of John's rating towards the item Apple. The rating value we can assume is $\delta = Volume$, that is, how many time John wrote in his blog or in his social network timeline about the item Apple. The average rating of all items is $\mu = 0.6$. Furthermore the topic Apple tends to be discussed more compared to the average topics, aproximately $b_i = 0.4$ more than average. On the other hand John is a user that write little and thus he has $b_u = -0.3$ lower than average. Thus the baseline estimate of b_{ui} for Apple by John is 0.7 by calculating the previous equation $0.6 - 0.3 + 0.4$.

$$\begin{array}{c} \\ John \\ Davide \\ Sara \end{array} \begin{array}{c} \begin{array}{ccc} \#Microsoft & \#Apple & \#Google \end{array} \\ \begin{pmatrix} 0.3 & ? & ? \\ ? & 0.4 & 0.4 \\ 0.1 & 0.6 & ? \end{pmatrix} \end{array} \tag{2.4}$$

Finally the bias has to be integrated into the general rating function:

$$min_{q^*,p^*,b^*} \sum_{(u,i)\in K} (r_{ui} - \mu - b_u - b_i - q_i^T p_u)^2 + \lambda(\left\|q_i\right\|^2 \left\|p_u\right\|^2 + b_u^2 + b_i^2) \tag{2.5}$$

To minimize the previous equation there are two most used approaches such as *stochastic gradient descent* [192] and *alternating least squares* (ALS) [179], the latter one also used and explained later in this work. Once solved the minimization, the recommender system completes this mapping and it can easily estimate the rating a user will give to any item by using Equation 2.1. Matrix Factorization techniques are nowadays largely used for recommendation task[119, 186, 194], thanks also to the large scale application that can be used into Big Data RSs [195, 1, 179].

2.1.5 Evaluation Measures

Even if the inputs of traditional recommendation and social recommendation are different, their outputs are the same, i.e., the predicted values for unknown ratings. Therefore, metrics that evaluate classic recommender systems can also be applied to evaluate social recommender systems. To evaluate RSs, the data is usually divided into two parts the training set K (known ratings) and the testing set U (unknown ratings). RSs will be trained based on K, and the quality of recommendation will be evaluated in U. Different evaluation metrics are proposed to evaluate the quality of recommendation from different perspectives, such as prediction accuracy, ranking accuracy, diversity and novelty, and coverage. Prediction accuracy and ranking accuracy are two widely adopted metrics.

Prediction Accuracy: Prediction accuracy measures the closeness of predicted ratings to the true ratings. Two widely used metrics in this category are Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE).

The metric RMSE is defined as:

$$RMSE = \sqrt{\frac{\sum_{u_i,u_v\in\upsilon}(R_{ij} - \hat{R_{ij}})^2}{|\upsilon|}}, \tag{2.6}$$

where $|\upsilon|$ is the size of υ and $\hat{R_{ij}}$ is the rating predicted from u_i to v_j.

The metric MAE is defined as:

$$MAE = \frac{1}{|\upsilon|} \sum_{u_i, u_v \in \upsilon} \left| R_{ij} - \hat{R}_{ij} \right|. \tag{2.7}$$

A smaller RMSE or MAE value means better performance, and due to their simplicity, RMSE and MAE are widely used in the evaluation of recommender systems.

Ranking Accuracy: Ranking accuracy evaluates how many recommended items are purchased by the user. Precision and recall are two popular metrics in this category. Recall captures how many of the acquired items are recommended, while precision captures how many recommended items are acquired, for example, *Prec@N* is used to indicate how many top-N recommended items are acquired. Long recommendation lists typically improve recall, while reducing precision. Therefore F-score is a metric combining them, and it is less dependent on the length of the recommendation list. Another popular metric is Discount Cumulative Gain (DCG), which is defined:

$$DCG = \frac{1}{|u|} \sum_{u_i \in u} \sum_{j=1}^{|L|} \frac{\hat{R_{i,j}}}{max(1, log_b j)} \tag{2.8}$$

where L is the ranked list of recommended items.

Finally is described one of most used evaluation metrics in this work, that is, Success at Rank k. The Success at Rank k (or S@k) is defined as the probability of finding a good recommendation among the top k recommended items. In other words, S@k is the percentage of runs in which there was at least one relevant item among the first k recommended items.

2.2 Social Network Analysis

Social networks have become very popular in recent years because of the increasing proliferation of Internet enabled devices such as personal computers, smartphones, and mobile devices. This is evidenced by the promising popularity of many online social networks such as *Twitter*, *Facebook*, and *LinkedIn*. Such social networks have lead to a tremendous explosion of network-centric data in a wide variety of scenarios. Social networks can be defined either in the context of systems such as *Facebook* which are explicitly designed for social interactions, or in terms of other sites such as *Flickr*[1] and *Instagram* which are designed for a different service such as content and media sharing, but which also allow an extensive level of social interaction. Twitter, is a particular social network and currently the most prominent microblogging service, serves more than 600 million users who post

[1] `www.flickr.com`

over 340 million short messages every day, sharing their thoughts, interests and activities with the public. On microblogging platforms, users are able to post messages, which are limited to a certain maximum length (e.g., 140 characters on Twitter), as well as share (*Retweet*) messages of other users. In addition, users can follow other users so that they can receive the latest posts published by those users. Microblogging services such as Twitter also provide APIs [2] that allow third parties to access microblogging data and develop various external applications such as systems for event detection [150], sentiment analysis [135] and recommender systems [81]. This specific social network is widely used in this work, thanks to the access to a public API and the huge numbers of free data available for research.

In general, a social network is defined as a network of interactions or relationships, where the nodes consist of actors, and the edges consist of the relationships or interactions between these actors. A generalization of the idea of social networks is that of information networks, in which the nodes could comprise either actors or entities, and the edges denote the relationships between them. Clearly, the concept of social networks is not restricted to the specific case of an internet-based social network such as *Facebook*; the problem of social networking has been studied often in the field of sociology in terms of generic interactions between any group of actors. Such interactions may be in any conventional or non-conventional form, whether they be face-to-face interactions, telecommunication interactions, email interactions or postal mail interactions.

An important clue about the structure of social networks came from a remarkable experiment by the American psychologist Stanley Milgram [128]. Milgram went out to test the common observation that no matter where we live, the world around us seems to be small: we routinely encounter persons not known to us who turn out to be the friends of our friends. Milgram thus not only wanted to test whether we are in fact all connected but he was also interested in what is the average distance between any two individuals in the social network of the American society. Milgram calculated the average of the length of the chains and concluded that the experiment showed that on average Americans are no more than six steps apart from each other, this was the source of the expression *six degrees of separation*. This is also referred to as the *small world phenomenon*. This phenomenon was tested in the context of MSN messenger data, and it was shown in [113] that the average path length between two MSN messenger users is 6.6. This can be considered a verification of the widely known rule of "six degrees of separation" in (generic) social networks.

The growth of online social networks raises many research topics that are analyzed in [10]:

[2] `https://dev.twitter.com/overview/api`

- *Statistical Analysis of Social Networks*: the important statistical properties of "typical" social networks, such clustering and connectivity behaviour.

- *Random Walks and their Applications in Social Networks*: ranking is one of the most well known methods in web search. Starting with the well known page-rank algorithm [6] for ranking web documents, the broad principle can also be applied for searching and ranking entities and actors in social networks.

- *Community Detection in Social Networks*: one of the most important problems and tasks in the context of social network analysis is the community detection, that is, the identification of group formation within a network.

- *Node Classification in Social Networks*: a large part of social network data can be modeled as labels associated with individuals, that can be in many forms: demographic labels, such as age, gender and location; labels which represent political or religious beliefs; labels that encode interests, hobbies, and affiliations; and many other possible characteristics capturing aspects of an individual's preferences or behavior. Given a social network with labels on some nodes, the task is to provide a high quality labeling for every node.

- *Evolution in Dynamic Social Networks*: this task studies how network vary over time when new nodes join or leave and new link emerge or delete. Furthermore is interesting how communities evolve and how its structure change.

- *Social Influence Analysis*: since social networks are primarily designed on the basis of the interactions between the different participants, it is natural that such interactions may lead to the different actors influencing one another in terms of their behavior.

- *Expert Discovery in Networks*: social networks can be used to discover influencer or expert in a specific task or topic.

- *Link Prediction in Social Networks*: one of the most important task is to discover how works the link formation between nodes, and also the prediction of it.

- *Privacy in Social Networks*: social networks contain various information about the individual in terms of their interests, demographic information, friendship link information, and other attributes. This can lead to disclosure of different kinds of information in the social network, such as identity disclosure, attribute disclosure, and linkage information disclosure.

- *Visualizing Social Networks*: as social networks became larger and complex, visualization techniques are growing faster to provide a natural way to summarize the information in order to make it much easier to understand and analyse.

- *Data Mining in Social Media*: social media provides a wealth of social network data, which can be mined in order to discover useful business applications. Data mining techniques provide researchers the tools needed to analyze large, complex, and frequently changing social media data.

- *Text Mining in Social Networks*: much of the recent researches are considering this topic due to the humongous textual content that can be found in social networks. From news articles to blog post, from UGC (*User-generated content*) to user profiles, text mining is one of the basic task for social network analysis. Furthermore, this topic is widely considered in almost all of the works presented in this thesis and refers also to the *Sentiment Analysis* task.

- *Integrating Sensors and Social Networks*: concern the integration of sensors such as mobile GPS and social media to improve the analysis in the specific context.

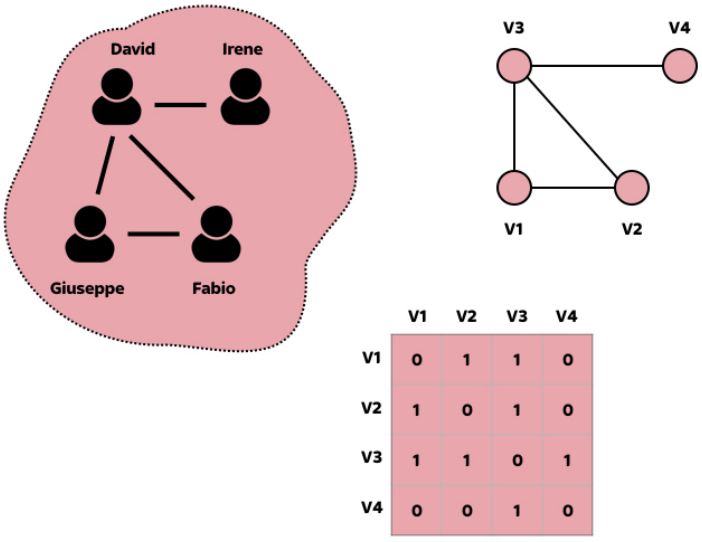
- *Multimedia Information in Social Media*: many popular social networks (*Youtube*, *Instagram*, *Flickr*) provide the opportunity to share media information such as videos, images, GIF. This kind of information is constantly growing, and such rich context-based information can be mined for a wide variety of applications by leveraging the combination of user behaviour and media data.

2.2.1 Definitions

Social Network Analysis has developed a set of concepts and methods specific to the analysis of social networks. In the following, we introduce the most basic notions of network analysis and the methods. For a complete reference to the field of social network analysis, we refer the reader to the exhaustive network analysis of Wasserman and Faust [176] or a more accessible introductory text on network analysis [155].

As discussed above, a (social) network can be represented as a graph $G = (V, E)$ where V denotes the finite set of vertices and E denotes a finite set of edges such that $E \subseteq V \times V$. Recall that each graph can be associated with its characteristic matrix $M := (m_{i,j})n * n$ where $n = |V|, m_{i,j} = \begin{Bmatrix} 1 & (v_i, v_j) \in E \\ 0 & otherwise \end{Bmatrix}$. Sometimes network analysis methods are easier to understand when we conceptualize graphs as matrices as proposed in Figure 2.2. Note that the matrix is symmetrical in case the edges are undirected. We will talk of a valued

graph when we are also given a real valued weight function $w(e)$ defined on the set of edges, i.e. $w(e) := E \times \mathbb{R}$. In case of a valued graph, the matrix is naturally defined as $m_{i,j} = \begin{Bmatrix} w(e) & (v_i, v_j) \in E \\ 0 & otherwise \end{Bmatrix}$. *Loops* are not excluded in the above definition, although they rarely occur in practical social network data sets. (In other words, the main diagonal of the matrix is usually empty.) Typically, we also assume that the network is connected, i.e. there is a single (weak) component in the graph. Otherwise we choose only one of the components for analysis.



	V1	V2	V3	V4
V1	0	1	1	0
V2	1	0	1	0
V3	1	1	0	1
V4	0	0	1	0

Figure 2.2: Network analysis works both on graphs and matrix representation

One of the pioneer of network structure was Milgram [128] with the identification of the *small world phenomenon*. One of the practical impact of his studies is that we can for sure exclude certain kind of network structure as possible models for social network. The three graph shown in Figure 2.3 is one of them. However, a tree is unrealistic because it shows no *clustering*: we all know from practice that our friends are likely to know each other as well because we tend to socialize in groups (if not for other reasons than other friends know each other because we introduced them to each other). Clustering for a single vertex can be measured by the actual number of the edges between the neighbors of a vertex divided by the possible number of edges between the neighbors. When taken the average over all vertices we get to the measure known as *clustering* coefficient. The clustering coefficient of a tree is zero, which is easy to see if we consider that there are no triangles of edges (triads) in the graph. In a tree, it would never be the case that our friends are friends with each other.

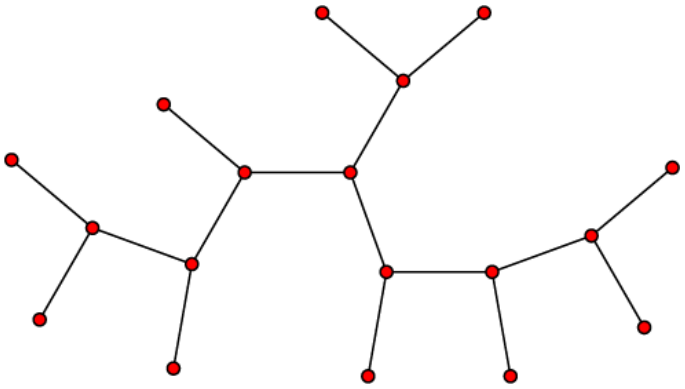


Figure 2.3: A simple representation of a Tree Graph.

The tree and other structure that are inapplicable to social networks, also have the rather unappealing characteristic that every node has the same number of connections. We know from our everyday walks in life that some of us have much larger social circles than others. The random graph model proposed by the Hungarian mathematicians Erdős–Rényi offers a new vision. A random graph can be generated by taking a set of vertices with no edges connecting them. Subsequently, edges are added by picking pairs of nodes with equal probability. This way we create a graph where each pair of vertices will be connected with an equal probability (this probability is a parameter of the process).

If we continue the process long enough - we choose a high enough probability - the resulting random graphs will have a small characteristic path length and most likely exhibit some clustering. (Needless to say if we go on we end up with a complete graph.) Still, we can raise significant concerns against the cold probabilistic logic of a random graph. Due to limitations of space -if not for other reasons - we are unlikely to make friends completely in random from anywhere in the world. Although we meet strangers occasionally by sitting next to them on an airplane, we mostly socialize in a given geographic area and even then in limited social environments such as our work and living space. Again, the friends of our friends are likely to be our friends as well. If that happens in a random graph, it happens by accident. Nevertheless, the Erdős–Rényi random graphs are interesting in the sense that they are examples of generative models. That is, random graphs are not (only) defined by what they are but also how they arise, i.e. the process of growing such a graph. These kinds of processes are also at the centerpoint of interest for the field of complex networks in physics where researchers study the emergence of complex global structures from systems that are defined solely through elementary interactions between primitive elements.

While alpha and beta models presented by the mathematicians Steven Strogatz and Duncan Watts [177] generate networks with small path lengths and relatively large clustering coefficients, they fail to represent an important feature of networks in nature: the scale-

free characteristic of the degree distribution. The understanding of this phenomenon and the construction of relative model is thanks to another Hungarian Albert-László Barabási.

To understand the scale-free phenomenon we have to look at the degree distribution of networks. Such a diagram shows how many nodes in the network have a certain number of neighbors (degrees). In a toroidal lattice all nodes have an equal number of neighbors. In the alpha and beta models as well as the random graphs of Erdős–Rényi this distribution is a normal distribution: there is an average degree, which is also the most common one. Degrees deviating from this are increasingly less likely. In real social networks, however, this distribution shows a different picture: the number of nodes with a certain degree is highest for small degree and the number of nodes with a given degree rapidly decreases for higher degrees. In other words, the higher the degree the least likely it is to occur. What is also surprising is the steepness of the distribution: the vast majority of the nodes have much fewer connections than the few hubs of the network. The exact correlation is a *power law* i.e. $p(d) = d^{-k}$ where $k > 0$ is a parameter of the distribution.

Barabási not only discovered that this is a fundamental characteristic of many networks that he studied, but also gave a generative model to reproduce it. The key of this model is that when adding new nodes we link the node to an already existing node with a probability that is determined by how many edges the node already has. In other words, a node that has already attracted more edges than others will have a larger probability to attract even more connections in subsequent steps. The works of Watts, Erdős–Rényi, Barabási and his colleagues are largely responsible for bringing network research into the scientific forefront. By analyzing those model instead of particular instances of it allows scientists to formulate precise and general claims about specific networks, from understanding the spread of viruses, to solving the vulnerability of a network attack, or creating more efficient structure for peer-to-peer networks. Furthermore this social network analysis fundamentals, are the basis concepts that raise the idea of the specific community detection algorithm that will be presented in Chapter 5.

2.2.2 Community Detection

The discovery of cohesive groups, cliques, and communities inside a network is one of the most studied topics in social network analysis. It has attracted many researchers in sociology, biology, computer science, physics, criminology, and many more. Community detection aims at finding clusters as sub-graphs within a given network. A community is then a cluster where many edges link nodes of the same group and few edges link nodes of different clusters. A general approach to community detection consists in considering the network as a static view in which all the nodes and links in the network are kept unchanged

throughout the study. Recent studies focus also on community evolution since most social networks tend to evolve over time through the addition and deletion of nodes and links. As a consequence, groups inside a network may expand or shrink and their members can move from one group to another one over time.

Most of the studies on community evolution use topological properties to identify the updated parts of the network and characterize the type of changes such as network shrinking, growing, splitting, and merging. However, recent work has focused on community evolution/detection by relying entirely on the behavior of group members in terms of the activities that occur in the network rather than exclusively considering links and network density. Other approach combines natural language processing together with social network analysis to explore Twitter messages in order to identify actionable ones, construct an actionable network, identify communities with their central actors, and show the behavior of the community members [170]. Many works uses community identification to understand communities, sub-communities and trends regarding political elections [162]. Finally, some approach use the *homophily* concept, that is, the tendency of individuals with similar characteristics or interests to associate with each other, to assess similarity between actors and the group homogeneity they have [147]. This work exploit how a group is cohesive in a social network, the cohesiveness of a group is a social factor that assesses how members of a group are close to each other, and may help predict a possible community splitting or disaggregation. This kind of approach raises new challenge and topic for community detection, that is, the *link prediction* task.

Link prediction is an important task in social network analysis, and aims at predicting if two given nodes have a relationship or will form one in the near future [115]. It is exploited in many social media applications such as the ones that need an embedded RSs to suggest new and relevant ties to the users. Like in community detection, similarity and proximity principles are widely used for link prediction. Moreover, information about network communities can improve the accuracy of similarity-based link prediction methods. Some works studies both supervised and unsupervised link prediction in networks where nodes may belong to more than one community, procreating different types of collaborations [174]. Other particular and recent approach studies user behavior of an online dating website in order to understand how user attributes can help predict who will date whom [181]. The task of link prediction is widely tackle in this thesis, where the link refers to a user, that is, link prediction as a user recommendation in social media. This specific RSs is recently studied in the literature and counts techniques for user recommendation on *Facebook*, *Twitter*, *Instagram*, *Tumblr*, *Reddit*, and all of the most used social media websites. Specifically some approach used *Latent Dirichlet Allocation* to discover

the formation of community on Twitter based on textual patterns [193], while [194] proposes the use of both LDA and MF to recommend users on Twitter and Chinese social network Weibo [3]. One of the works presented in this thesis at Chapter 5, leverages clustering techniques to identify specific communities in order to improve the user recommendation precision.

Informally, a community in a network is a group of nodes with greater ties internally than to the rest of the network. This intuitive definition has been formalized in a number of competing ways, usually by way of a quality function, which quantifies the goodness of a given division of the network into communities. Some of these quality metrics, such as Normalized Cuts [156] and Modularity [131] are more popular than others, but none has gained universal acceptance since no single metric is applicable in all situations. Algorithms for community discovery vary on a number of important dimensions, including their approach to the problem as well as their performance characteristics. An important dimension on which algorithms vary in their approaches is whether or not they explicitly optimize a specific quality metric. Spectral methods [172], the Kernighan-Lin algorithm [103] and flow-based post-processing [51] are all examples of algorithms which explicitly try to optimize a specific quality metric, while other algorithms, such as Markov Clustering (MCL) [171] and clustering via shingling do not do so. Another dimension on which algorithms vary is when they let the user control the granularity of the division of the network into communities. Some algorithms (such as spectral methods) are mainly meant for bi-partitioning the network, but this can be used to recursively subdivide the network into as many communities as desired. Other algorithms such as agglomerative clustering or MCL allow the user to indirectly control the granularity of the output communities through certain parameters. Still other algorithms, such as certain algorithms optimizing the Modularity function, do not allow (or require) the user to control the output number of communities at all. Coming to performance characteristics, algorithms also vary in their scalability to big networks, and multi-level clustering algorithms provide a powerful framework for fast and high-quality graph partitioning such as Metis [101], MLR-MCL [153] and Graclus [46] and local clustering algorithms [160] that scale better than many other approaches.

2.2.3 Social Networks Research Topics

Social network analysis and social mining can be very useful in this context where RSs can take benefit from social networks and conversely, where the formation and evolution of the

[3] `http://weibo.com`

network can be affected by the recommendations. In order to illustrate this point, we can mention three well known tasks in social network analysis and social network mining:

The first one is the identification of key actors which play a particular role or which have a particular position in the network. Different indicators, such as the centrality or the prestige were initially introduced mainly in order to highlight the "most important" actors in the network [176]. With the appearance of online social networking, these measures were recently revisited to detect actors called, depending on the authors, mediators, ambassadors or experts. Among the actors who have received a lot of attention appears notably the influencer who can be defined as an actor who has the ability to influence the behaviour or opinions of the other members in the social network [13]. The identification of the influencers can be seen as an optimization problem better known as "influence maximization" (or "spread maximization") that is NP-complete but approximated solutions can be determined thanks to greedy algorithms like "Cost-Effective Lazy Forward" (CELF) algorithm or its extensions Newgreedy, Mixedgreedy or Celf++ [102].

Another well known problem in the context of social networks is that of community detection. This problem has mainly been studied in the literature in the case where the community structure is described by a partition of the network actors where each actor belongs to one community [171, 52] and among the core methods we can mention those that optimize a quality function to evaluate the goodness of a given partition, like the modularity, the ratio cut, the min-max cut, or the normalized cut, the hierarchical techniques like divisive algorithms based on the minimum cut, spectral methods or Markov Clustering algorithm and its extensions. However, in real networks, an actor can often belong to several groups and these overlapping communities can be detected using for example the clique percolation algorithm implemented in CFinder or OSLOM (Order Statistics Local Optimization Method). Other recent works have attempted to detect communities, taking into account the profile of the users and their relationships [193]. These methods can be applied to determine groups of users with similar characteristics or the same interests and consequently, they can be integrated in collaborative RSs.

The evolution of the network is another challenge. Indeed, in many networks, the structure of the network, in other word the actors as well as their relationships, changes quickly over time. The identification of evolving communities or their detection over time is also a subject of recent research which can be integrated in systems to improve recommendations but the dynamic analysis of the network is also related to the link prediction problem which aims to determine the appearance of new links or the deletion of links in the network [62, 115, 12]. It is obvious that link prediction can be useful for people recommendation and, conversely, recommendation approaches can allow to predict the evolution

of the network. This temporal dimension is notably important in the context of mobile applications in which moving actors are interacting with each other.

2.3 Sentiment Analysis

In this chapter is described the role of Sentiment Analysis in this research and in more general scenario. Will be defined properties and taxonomies of sentiments giving examples and definitions. In addiction will be presented some state-of-the-arts methodologies for sentiment classifications and open research problems.

2.3.1 Definitions

Sentiment Analysis is an Artificial Intelligence and NLP field that have the aim to determine the presence of subjectivity within expressions. For simplicity we can assume that this expressions are always textual phrases, but there are various studies that refer to sentiment analysis of videos or images [197] [92].

The following definition, from the survey proposed by Liu [116], results clear and simple: Sentiment Analysis studies the opinions, sentiments and the emotions expressed in a text.

Starting from this definition, is possible to note that this three concept are slightly different, and this will be discussed forward in this chapter. To better understand the next definitions, we can refer to the text example below:

"'(1) I bought an iPhone a few days ago. (2) It was such a nice phone. (3) The touch screen was really cool. (4) The voice quality was clear too. (5) Although the battery life was not long, that is ok for me. (6) However, my mother was mad with me as I did not tell her before I bought it. (7) She also thought the phone was too expensive, and wanted me to return it to the shop. ... "' [116]

Reading the above examples, we can easily argue how is possible to convey opinions in really different ways. First of all is possible to note that some opinions are positive (2,3 and 4), other negative (5,6, and 7), and still other do not contains opinions (1). Secondly it is noted that the subject of the comment frequently change, passing from "iPhone" in general (1) to some of its features that is touch screen (3), sounds quality (4), battery lifetime (5), and price (7).

Now it is possible to introduce some important concept, that is, the Object or Entity.

Object or Entity: is the entity on which a comment is expressed.
Entity concept has been classified by Liu [116] listing exactly five types: *product, person,*

Entity Type	Example
Product	I'm really loving my *iPhone* #loveit
Person	*Tiger Woods* was simply amazing yesterday!!
Organization	I will never give any chance to *Samsung*
Event	Last *Superbowl* was one of the best sports event ever!
Topic	Is irritating when something is happening and you can't control it!

Table 2.2: Taxonomy of the entities on which are expressed opinions or sentiments

organization, event and topic. This taxonomy is resumed in table 2.2 and even if simple it is quite complete.

The Entity, the *iPhone* in the previous example, has some detailed components *battery,touch screen, color, price* and many more. All of this kind of attributes is included in the concept of Feature:

Feature: represent an attribute, a component of the Object or Entity

Going in details regarding the features is possible to also define that exists different kind of feature [116]:

Explicit Feature: it is when the feature, or a synonym, explicitly appear in text **Implicit Feature:** it is when the feature, or a synonym, does not explicitly appear in text or is implied

Giving an example, in the phrase "'*this phone has an exorbitant price!*"' the feature *price* appear explicitly in text. On the contrary in the phrase "'*this phone is too expensive!*, the presence of feature price is implied by the word *expensive*. An opinion regarding a feature can be mainly expressed in two ways: comparative or direct.

A **Comparative Opinion:** express a similarity, a difference between two Objects or between two Features.
The study of this comparative expressions is quite complex and is an open research problem M. Ganapathibhotla e Bing Liu [59].

In the following definition, it is described a definition about the parts that compose a direct opinion:

Direct Opinion: it is a quintuple (**o, f, oo, h, t**), where **o** is the Object, **f** is the Feature of the Object o, **oo** is the polarity of the expressed opinion about the feature **f** regarding the Object. **h** is the opinion-holder and **t** is the time when the opinion is expressed by the

opinion holder **h**.

The **Polarity**, indicates whether an opinion is **positive**, **negative** or **neutral**. The **opinion-holder** is the subject that comments on feature f, and thus is the one who expresses an opinion with positive or negative polarity. Taking into account of the previous example *"She also thought the phone was too expensive"*, the opinion-holder is she, that is, the person that expresses the opinion regarding the price. Recognising the opinion-holders, in addiction to being hard, it's not so interesting for sentiment analysis general purposes.

Different concept is about *Subjectivity* and *Sentiment*. This two concept may appear quite similar, but in the literature are studied as two different classification problem [138]. The **Subjectivity Classification** solves if a text or a phrase contains an opinion or not, and a non subjectivity text can be labeled as *Objective* text. On the contrary the **Sentiment Classification** determines if a subjective text is positive or negative. Recovering the previous example we can now classify the phrase *"'I bought an iPhone a few days ago"'* as not subjective or objective and (2) *"'It was such a nice phone"'* as an subjective phrase and specifically a positive phrase.

The problem of sentiment classification is sometimes treated as a binary problem, as we discussed since now, to identify positive or negative sentence. Many state-of-the-arts approaches such as [43, 108] consider the sentiment classification as a multi-variate classification problem, in which the final classification is not limited to positive or negative but can also identify much more deeply the several scale of emotions and sentiment polarities as illustrated into the Plutchik emotion wheel in Figure 2.4. Since now, we used to see many different scale of products' review, from 1 to 5 stars or from 1 to 10 approval, but if having a multi-variate scale of sentiment is an index of completeness, on the other hand this can result much more hard to achieve and also inaccurate. Furthermore have to take into consideration that can be more easier to give a two way opinion compared to giving ten point scale opinion, and also in the latter case the opinion of two people can be a lot different. This topic will be resumed in the following section when will be discussed some sentiment classification algorithms.

2.3.2 Machine Learning and other approaches

In this section are described most used methodologies for sentiment analysis classification, best practices from state-of-the-arts techniques, and useful corpus and resources to use into sentiment analysis algorithms.

Most of the current sentiment analysis techniques can be referred to machine learning techniques [138, 139, 136], where two classes are considered positive or negative, that is,

Figure 2.4: The Plutchik emotion wheel

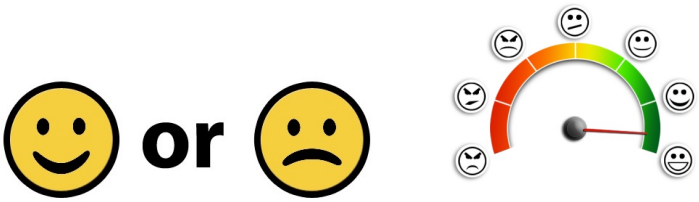


Figure 2.5: Binary sentiment and multiple sentiment

sentiment classification as already defined in previous sections.

Supervised Learning. Supervised learning is the machine learning task of inferring a function from labeled training data. The training data consist of a set of training examples, that is, in the scenario of sentiment analysis a list of phrase with the labeled polarity. We can therefore define:

- Input data as a set I (typically vectors)
- The set of output data O. Output can be a regression or a number.
- A function h that combine at each input data (I) the corresponding right output (O)

Any supervised learning algorithm is based on the assumption that having a relevant number of labeled examples, the algorithm will be able to create a function f that will approximate function h. If the approximation of h will results appropriate, when new input data will be proposed to f, the function might will be able to provide a correct output answer similar to answer provided from function h. This kind of learning is typically fast and precise, despite of problems like *overfitting* [94] and the requirement to have a large number of labeled data on input.

Unsupervised Learning. Unsupervised Learning consists of a class of problems in which trying to solve automatically how data are organized. During the learning phase are provided only non-labeled examples, as the class are unknown but have to be learned automatically. Unsupervised approaches to sentiment classification can solve the problem of domain dependency and reduce the need for annotated training data. Turney [168] uses two arbitrary seed words (poor and excellent) to calculate the semantic orientation of phrases, where the orientation of a phrase is defined as the difference of its association with each of the seed words (as measured by pointwise mutual information). The sentiment of a document is calculated as the average semantic orientation of all such phrases. Unsupervised learning techniques generally works comparing or clustering data finding similarity or difference patterns in the data.

Classification algorithms are used in both supervised and unsupervised learning. The most used algorithms are Support Vector Machines SVM [139, 110] in the top ten of most used algorithms in data mining [180], Naive Bayes [53, 30], Neural Networks [64], and also clustering algorithms such as K-means, and matrix factorization [89]. Some research also showed how is it possible to create proprietary algorithms [58] to improve classification precision.

Any of those approaches surely have to analyze the terms contained in text, and the most frequent techniques are the use of *bag-of-words* and *N-Grams*. The first one consists

of representing the number of co-occurrence of a specific term in a entire analyzed document. In this way is possible to take into account of the term frequency, but completely loose the grammar, and therefore the context of the text. The second technique uses n-grams, that is, a list of N consecutive words extracted from a phrase. Basing of the value of N, is it possible to assume different analysis of sentiment. Usually the value of N is between two (bigram) and three (trigram). The most recent Natural Language Processing techniques are currently working on language models realized from N-grams that make Markov chains [37]. These approaches analyzes the frequency of specific terms and infer on texts to determine sentiment polarity with techniques from Information Retrieval such as **tf-idf**. Other fundamental methodology of text analysis is the use of **POS Tagger**. The goal of POS tagger, namely Part of speech tagger, is to discover the grammar kind of a word. Giving in input at Stanford POS tagger [4] the sentence *"'I am a good boy"'* is obtained in output:

"'I|PRP am|VBP a|DT good|JJ boy|NN"'

As can be noted, at each word corresponds a symbol that means the grammar form in which it appertain. Referring to table 2.3 are resumed the most used symbol for Stanford POS tagger, the tagger that is used in this work.

POS tagging is an important task for sentiment analysis because one of the most used techniques concern the isolation of specific terms and specific term pattern with high probability of polarity such as adjectives and adverbs. Some other approaches used POS tagged words to train machine learning classifier and improve the sentiment classification [168].

2.3.3 Algorithmic Approaches

In this section are explained methods that do not use machine learning techniques, but classify sentiment analyzing every single word within a sentence. Words that shape a sentence in positive or negative way, are known in literature as *opinion words* or *polar words*. Positive opinion words are for example *good*, *"'excellent"'*, while *"'awful"'*, and *"'ugly"'* represent negative opinion words. Comparative words such as *"'better"'* and *"'worst"'* don't belong to the opinion word dictionary, but have to be treated in different ways.

The task of opinion word retrieval, or understand how to automatically identify those words, can be considered one of the hardest task in sentiment analysis. One of the problem is because opinion words are topic and domain dependent, for example the opinion word

[4] `http://nlp.stanford.edu/software/tagger.shtml`

Pos Tag	Description	Example
CC	conjunction	and
CD	cardinal numbers	1, third
DT	def article	the
EX	there esistenziale	*there* is
FW	unknwown word	lol, asd
IN	preposition	in, of, like
JJ	adjective	good
JJR	comparative adjective	better
JJS	superlative adjective	best
LS	lists	1)
MD	modal	could, will
NN	name, singolar	table
NNS	name plural	tables
NNP	noun, proper, singular	David
NNPS	noun, proper, plural	Vikings
POS	genitive marker	friend's
PRP	pronoun, personal	I, he, it
PRP$	pronoun, possessive	my, his
RB	adverb	however, usual, naturally
RBR	adverb, comparative	better
RBS	adverb, superlative	best
RP	particle	give up
TO	to	*to* go, *to* leave
UH	interjection	ooohh, uuhhh, ahahah
VB	verb, base form	take
VBD	verb, past tense	took
VBG	verb, present participle or gerund	taking
VBN	verb, past participle	taken
VBP	verb, present tense, not 3rd pers singular	take
VBZ	verb, present tense, 3rd pers singular	takes
WDT	WH-determiner	which
WP	WH-pronoun	who, what
WP$	WH-pronoun, possessive	whose
WRB	WH-adverb	where, when

Table 2.3: Treebank Pos Tagger List - Stanford University

Table 2.4: A comparison of different algorithmic approach.

Feature	Manual	Dictionary	Corpus
Automation	▼▼	▲	▲▲
Costruction Time	▼▼	▲	▲
Rules Complexity	▲	▼	▼
Precision	▲▲	▲	▲
Completeness	▲	▲	▼
Domain Dependent	▲	▼▼	▲▲
Term Recall	▲	▲▲	▼

"'long"' in the context of *battery life* has a positive polarity. The same word referred to *queue* has a negative polarity. Since this kind of example can be unlimited, determine a list of positive or negative words can be really hard. A baseline approach to collect and manage an opinion word set is to collect manually the largest number of words that matches the specific domain is needed. This manual approach is of course the most precise, but it is not feasible. For this reason is nearly always used in combination with automatic techniques.

Dictionary-based approaches represent an example of semiautomatic techniques, since use a starting set of opinion words *seed set* that is manually retrieved [88]. Starting from a small opinion word set of known polarity, is possible to expand this set with synonyms and contraries found on Wordnet [129]. Once the process is completed, a domain expert can correct or remove some words from the positive or negative set. Another common strategy is to generate additional information to the created set, by integrating a glossary or the use of machine learning algorithm [48]. The biggest problem using *dictionary-based* techniques is to identify opinion word with a specific polarity for a certain domain.

Corpus-based methods are based on the identification of recurring pattern within the textual sentences. This type of approaches corpus-based since require for the learning process a huge numbers of textual examples. One of this methods [98] uses a seed set of adjectives and grammar ties about most used conjunctions (*and, or, but,*), in order to identify other adjectives that express a specific polarity. To give an example starting from the phrase *"This car is beautiful and spacious"*, if is note that *beautiful* is a positive opinion word, using the conjunction and we can assert that also spacious is a positive opinion word. Looking at an entire corpus is thus possible to create an adjective graph when a tie is the connection between two adjective using conjunction, and therefore the final sentiment classification is easy.

In Table 2.4 are compared the described methodologies. The approaches that are based on corpus are not effective as the one based on dictionaries regarding the *recall* of opinion

words. This because is harder to find in a corpus all of possible terms that can be represented in a dictionary. The recall measure is used in *Information Retrieval* and indicates the proportion of relevant document retrieved. This is the formal definition of recall:

$$recall = \frac{Number\,of\,relevant\,documents\,retrieved}{Total\,number\,of\,relevant\,documents} \tag{2.9}$$

An exhaustive resume of main techniques of sentiment analysis is given by Prabowo and Thelwall [145] in Figure 2.6 and counts the most used and important state-of-the-arts techniques.

Turney [168] uses an unsupervised learning algorithm to classify reviews. The classification is binary, that is, the goal of the algorithm is to provide whether a review is suggested or not. The algorithm takes in input the text of the review and gives in output the binary classification following these steps:

1. extract sentences that contains adjectives or adverbs using a POS tagger,
2. compute the semantic polarity of each sentence,
3. classify the review obtaining the average polarity of each sentence.

To compute the polarity, are extracted couple of words that respect one of pattern showed in Figure 2.7, and the polarities are calculated verifying the nearness of these couple of words with two extreme sentiment as *poor* and *excellent*. The exact equation is given as follows:

$$\log\left(\frac{hits(fraseNEAR"excellent")hits("poor")}{hits(fraseNEAR"poor")hits("excellent")}\right) \tag{2.10}$$

where hits value is computed with a search engine like Altavista Advanced Search Engine[5] using the operator near in the research.

A different approach proposed by Pang and Lee [139] takes inspiration from Turney but investigating how the sentiment classification can be take back to a problem of *topic classification* where the two admitted classes are positive and negative. The evaluation tests are obtained using an IMDB corpus [6], where they manually tagged 700 positive and

[5] http://www.altavista.com/web/adv now yahoo.com

[6] `http://www.imdb.com`

S.No	Authors	Techniques	Features	Data Source	Results
1	Pang-Lee (2002)	NB, SVM, MaxEnt	Term Presence, Unigram, Bigram, Adjectives, POS Tagger	Movie Reviews	Best: SVM 82.9%
2	Koppel and Schler (2006)	NB,SVM, J4.8, Linear Reg	Bag of words, neutral class help classify pos/neg better?	Tv Corpus, Shopping.com Corpus	5 Fold-Cross Validation: best Linear regression.
3	Melville – Gryc – Lawrence (2009)	Multinomial Naïve Bayes,	Lexical Resources, Feature reduction	Lotus Blog, Political Blog	82% accuracy
4	Go – Bayani (2009)	NB, SVM, MaxEnt	Unigram, Bigram, POS Tagger	Twitter Stanford Corpus	82.70%
5	Prabowo (2009)	Hybrid: SVM, Rule-based, Statistic-based	Feature Selection (MI, Chi-square etc)	Movie Reviews(pang-lee), Myspace comments	Best approach: RB + SB + SVM
6	Davidov – Tsur (2010)	SVM, kNN (classify hashtag)	Pattern-based, hashtags and emoticons, n-grams	Twitter Corpus(pos/neg)	Cross-fold: 0.8 hash 0.86 smiley
7	Pak – Paroubek (2010)	NB	Unigram, Bigram, Trigram. Feature Reduction (user, url, stop words)	Twitter Training: 300.000 (Pos/neg/neutral) Test 216	Best: Bigram+Neg+salience
8	Barbosa – Feng (2010)	SVM, others	N-grams, POS tag, Tweet features (RT, smiley, etc.)	Twitter Sentiment, Twendz, TweetFeel (pos/neg/neutral)	Train: 2000. 81%
9	Frank-Bifet (2010)	NB, SGD, Hoeffding Tree	Feature Reduction, Text Norm, Repeated Letters	Twitter Edimburgh Corpus	86.26%
10	Kaiquan Xu (2011)	Multiclass SVM	Linguistic Feature	Amazon Reviews	61%
11	Rui Xia (2011)	NB, SVM, MaxEnt	Unigram, Bigram, dependency grammar	Movie reviews, multi-domain dataset	Best: SVM 86%
12	Microsoft Reasearch Asia (2011)	SVM light	Target Dependent, Stemming, Text Norm, POS Tagger	Twitter size 2000. Test 500 Manual (pos/neg)	Best: 85.6% target-dependent feature
13	Wilson-Moore-Koulompis (2011)	SVM	MPQA, POS Tag, Hashtag and Emoticons	Twitter Corpus (pos/neg/neutral)	Cross-fold: 0.74% (hash+emot)
14	Agarwal et al. (2011)	SVM	Text feature(#negation word, #positive word, SUM prior polarity etc.)	Twitter corpus Manual Labeled	-
15	Lee – Tan – Tang – Jiang et al. (2011)	SVM, kNN (classify hashtag)	User level SA, exploiting social relationship (following, mentions)	Twitter Corpus(pos/neg)	social rel. overcomes text classification
16	Saif – He – Alani (2012)	NB, MaxEnt	Semantic concepts (OpenCalais, Zemanta, etc.), Information Gain	Twitter Stanford Corpus	NB and Semantic Feat: 86.3% (+4% Go – Bayani)
17	Kontopoulos et al. (2013)	OpenDover (tool)	Using Ontologies to takle jargon and noisy tweet	Twitter Corpus (annotated entities)	-
18	Hu et al. (2013)	Least Square, Lasso	Soc. Rel: Sentiment consistency and emotional contagio	Twitter Stanford Corpus + users	social rel. overcomes text classification

Figure 2.6: Resume of main state-of-the-arts approaches for sentiment analysis classification

	First word	Second word	Third word (Not Extracted)
1.	JJ	NN or NNS	anything
2.	RB, RBR, or RBS	JJ	not NN nor NNS
3.	JJ	JJ	not NN nor NNS
4.	NN or NNS	JJ	not NN nor NNS
5.	RB, RBR, or RBS	VB, VBD, VBN, or VBG	anything

Figure 2.7: Two word pattern used by Turney [168]

700 negative reviews and evaluate the classifier with different algorithms: *Naïve Bayes*, *Maximum Entropy* and *Support Vector Machine*. Results are showed in Figure 2.8 and shows how efficient is *Support Vector Machine* compared with other algorithms.

	Features	# of features	frequency or presence?	NB	ME	SVM
(1)	unigrams	16165	freq.	**78.7**	N/A	72.8
(2)	unigrams	"	pres.	81.0	80.4	**82.9**
(3)	unigrams+bigrams	32330	pres.	80.6	80.8	**82.7**
(4)	bigrams	16165	pres.	77.3	**77.4**	77.1
(5)	unigrams+POS	16695	pres.	81.5	80.4	**81.9**
(6)	adjectives	2633	pres.	77.0	**77.7**	75.1
(7)	top 2633 unigrams	2633	pres.	80.3	81.0	**81.4**
(8)	unigrams+position	22430	pres.	81.0	80.1	**81.6**

Figure 2.8: Experimental results discovered by by Pang and Lee [139]

2.3.4 Corpus and Dataset

Nowadays the increasing interests of intelligent systems, and the use of machine learning techniques for natural language processing and information retrieval tasks, have raised the demand of dataset and corpus for testing and comparing algorithms. More concretely, this datasets contain dictionaries of positive and negative terms, or corpus with labeled data for supervised learning algorithms. For the only goal of information retrieval, this information can be extracted also from websites that provide a large number of different reviews such as *Amazon*[7], *Epinions*[8] or *ConsumerSearch*[9], or in addiction more specialized website such as *Rotten Tomatoes*[10], or *TripAdvisor*[11]. Most of this resources are available for English, but it

[7] http://www.amazon.com/
[8] http://www.epinions.com/
[9] http://www.consumersearch.com/
[10] http://www.rottentomatoes.com/
[11] http://www.tripadvisor.com/

is increasing the number of italian resources such as AlaTest[12] and Ciao.it[13]. Furthermore also social networks are really a great source of textual information both for sentiment analysis and information retrieval. In particular the most used is *Facebook* but has not any public API access to gather user data. The second one is *Twitter*, much more important for researcher because has a public API to gather textual data and users' information. This is also the main social network used in all of presented work in this thesis. Finally there are also other minor social network like *Instagram* and *Linkedin* but are much less interesting for sentiment analysis task.

Despite the large number of website and interesting social networks, there aren't many labeled dataset to use and especially for social networks and, more important, currently there are any standard social dataset to make some test. Below is possible to find some of this labeled datasets:

- **Cornell movie-reviews datasets**[14]: is composed from three datasets. The first one with labeled binary sentiment such as positive and negative. The second one with a range of sentiment between one and five, and the latter containing 5000 objective sentences and 5000 subjective sentences.

- **Movie Review dataset**: available online[15] is built by 25,000 movie reviews for training and 25,000 for testing. CITE ->

- **NTCIR multilingual corpus**: requires the registration[16] and also support Chinese and Japanese. Note that includes also who is the opinion holder within each sentence.

- **Stanford Twitter sentiment corpus** [17] consists of two different sets, training and test. The training set contains 1.6 million tweets automatically labelled as positive or negative based on emotions. For example, a tweet is labelled as positive if it contains :), :-), :), :D, or =) and is labelled as negative if it contains :(, :-(, or :(.

- **Obama-McCain 2008 Debate** [18] The dataset was constructed from 3,238 tweets crawled during the first U.S. presidential TV debate in September 2008 and tagged using Amazon Mechanical Turk and sentiment was tagged as positive, negative, mixed, or other.

[12] http://www.alatest.it
[13] http://www.ciao.it
[14] http://www.cs.cornell.edu/people/pabo/movie-review-data/
[15] http://ai.stanford.edu/~amaas/data/sentiment/
[16] http://research.nii.ac.jp/ntcir/permission/ntcir-7/ntcir7moat-xinhua-EandSC.html
[17] http://help.sentiment140.com/
[18] https://bitbucket.org/speriosu/updown

- **Sentiment Strength Twitter Dataset** [19] This dataset consists of 4,200 tweets manually labeled in a range of sentiment strength between -1 and +5.

- **Sentiment Strength Twitter Dataset** [20] The Sanders dataset consists of 5,512 tweets on four different topics (Apple, Google, Microsoft, Twitter). Each tweet was manually labelled as positive, negative, neutral, or irrelevant with respect to the topic.

- **Sentiment Strength Twitter Dataset** [21] This is a collection of datasets that were built for a challenge where participants evaluate their systems for sentiment analysis classification and many other tasks.

- **MPQA corpus**[178]: is available online [22], and contains more than 500 hundred news articles from different sources and manually labeled.

In addiction to the aforementioned dataset, other useful resources for sentiment analysis are dictionaries and glossaries. Below are showed the most used in the field of sentiment analysis:

- **Emotion Words**: This is a collection of adjectives divided by the type of sentiment. It can be found online at [23] and it is manually labeled.

- **The Linguistic Inquiry and Word Count (LIWC)**[24]: This is a software that helps to produce linguistic analysis of textual content. It refers to 2,290 unique linguistic stem such as if a word contains positive or negative emotion or if represents anger or stress and much more. In addiction, it is possible to extract the word dictionary that run the software.

- **Subjectivity Lexicon**: this resource is available to download at [25]. It contains a list of subjectives terms, the prior polarity of each term such as positive, negative, or neutral and also the strength of polarity if strong or weak. Is used by Wilson et al. [178].

[19] http://sentistrength.wlv.ac.uk/
[20] http://www.sananalytics.com/lab/
[21] http://alt.qcri.org/semeval2016/index.php?id=tasks
[22] http://mpqa.cs.pitt.edu/corpora/mpqa_corpus/
[23] http://www.derose.net/steve/resources/emotionwords/ewords.html
[24] http://www.liwc.net/
[25] http://mpqa.cs.pitt.edu/lexicons/subj_lexicon/

- **SentiWordnet**: it is also available online [26]. This is a lexicon created by the Institute of Science and Technology CNR[27] to support opinion mining techniques and algorithms. Infers on terms gathered from Wordnet and grouped by synonyms. For each term is defined a score from zero to one respecting of three parameters: positivity, negativity and neutrality.

- **WordNet Affect**: is available both for research and commercial use at [28]. It uses Wordnet to creare dictionaries for specific domain such as politics, economics, sports.

- **Affective Norms for English Words(ANEW)**[29]: this resource contains a set of normative emotional ratings for a large number of words in the English language. This set of verbal materials have been rated in terms of pleasure, arousal, and dominance in order to create a standard for use in studies of emotion and attention.

[26] http://sentiwordnet.isti.cnr.it/
[27] http://www.isti.cnr.it/
[28] http://wndomains.fbk.eu/wnaffect.html
[29] http://csea.phhp.ufl.edu/media/anewmessage.html

Chapter 3

Sentiment-based User Recommender on Twitter

Nowadays, the emerging popularity of Social Web raises new application areas for recommender systems. The aim of a social user recommendation is to suggest new friends having similar interests. In order to identify such interests, current recommender algorithms exploit social network information or the similarity of user-generated content. The rationale of this work is that users may share similar interests but have different opinions on them. As a result, considering the contribution of user sentiments can yield benefits in recommending possible friends to follow. In this paper we propose a user recommendation technique based on a novel weighting function, we named *sentiment-volume-objectivity (SVO)* function, which takes into account not only user interests, but also his sentiments. Such function allows us to build richer user profiles to employ in the recommendation process than other content-based approaches. Preliminary results based on a comparative analysis show the benefits of the advanced approach in comparison with some state-of-the-art user recommender systems.

3.1 Introduction

The growing popularity of social networks increases the availability of user sentiments, which has become a significant impact factor on buying decisions, brand reputations and public opinions. Furthermore, recommending pertinent news stories, documents, and users to follow, has long been a favourite domain for recommender systems research. Several new approaches harness real-time micro-blogging activity from services, such as Twitter[1], as the basis for identifying user preferences and filtering relevant contents to specific people. Recently, Twitter has become an interesting source of research activity as a result of the

[1] `twitter.com`

large amount of available user-generated data. In particular Twitter permits users to share a sentence - called tweet - to the followers, with a maximum length of 140 characters.

In this instance, the purpose of user recommendation is to identify relevant people to follow among millions of users that interact in the social network. Previous attempts include both content-based and graph-based approaches. The former focuses on metrics for measuring the topic similarity among Twitter users, the latter exploits the graph of relationships among users to infer correlations.

The main idea behind this work is that users may share similar interests but have different opinions about them. Therefore, we extend the content-based recommendation by means of the sentiments and opinions extracted from the user micro-posts in order to improve the accuracy of the suggestions. This leads us to define a novel weighting function in order to enrich content-based user profiles.

3.2 Related Work

In spite of the growing body of research on exploiting user-generated contents in recommendation engines, there are few attemps to consider sentiment included in micro-posts during the recommendation process. Singh *et al.* [159] introduce a hybrid recommender system that improves the results of collaborative filtering by incorporating a sentiment classifier in the movie recommendation scenario. Bank and Franke [21] try to better represent public product reviews on weblogs through different text mining techniques. Faridani [49] achieves the same goal by exploiting a multivariate regression approach. As far as we are aware, there are no attempts towards sentiment user recommendation in social networks.

User recommendation approches that ignore user opinions have been proposed by Freyne *et al.* [54] and Chen *et al.* [33] exploring different recommendations strategies. Guy *et al.* [76] propose a people recommendation engine within an enterprise social network site scenario. They aggregate several different sources to derive factors that might influence the similarity measure. Twittomender [84] lets users find pertinent profiles on Twitter exploiting different strategies, both content-based and collaborative ones. Arru *et al.* [15] propose a signal-based representation of user interests in order to draw similarities among people.

3.3 Sentiment Analysis Algorithm

Sentiment analysis or opinion mining is formally defined as the computational study of sentiments and opinions about an entity expressed in a text. According to Liu [116], the entity is classified into five categories: *product, person, brand, event, concept*. Particularly,

Table 3.1: Emoticons Noisy Label

Positive	Negative	Neutral
:)	:(	:|
:-)	:-(	-.-
:D	;(	
;)	;-(	

in this work we assume the *concept* as the sentiment analysis target entity. Sentiment analysis is a difficult task, hence - before the setup of the algorithm - some assumptions are needed. There are multiple granularity levels of sentiment analysis, as explained in [9]: feature-level, entity-level, sentence-level, document-level.

In this work we consider sentiment analysis at sentence-level. Specifically, in the Twitter domain we assume that a sentence matches the whole tweet. Moreover, we assume that each sentence contains only one opinion related to the entity.

The goal of our sentiment analysis system is to obtain an output value that represents how much positive, negative or neutral is the sentiment expressed in a tweet. For this reason, we implemented a Supervised Machine Learning algorithm based on a Naïve Bayes classifier. With a view to training our algorithm, we needed a dataset with labeled tweets. However, due to the lack of a Twitter public dataset, we decided to follow an alternative approach. Instead of manually building a labeled dataset, Bhayani *et al.* [65] propose to employ a noisy dataset of positive, negative, and neutral tweets. The labels correspond to special sequences of characters in the tweets, such as positive or negative emoticons (e.g., `:-D` `;-(`), hashtags (e.g., `#iloveit`, `#ihate`) or keywords (e.g., `good`, `sad`). Even though these labels do not always correspond to the right sentiment expressed by the tweet, they allow us to collect a large amount of data for training.

The Twitter APIs[2] have been used to retrieve a set of tweets containing the aforementioned features. The final training dataset counts 150000 tweets divided in 50000 tweets for each class. Because the experimental evaluation is conducted on events related to the 2013 Italian political elections, the TextCat language recognizer[3] is employed to limit the set to Italian tweets. In order to increase the classifier precision and reduce the presence of noise, we performed a feature selection. In particular, the terms with low values of *Salience* are discarded. The *Salience* of a term t is defined by Pak et al. [135] as follows:

$$Salience(t) = \frac{1}{N} \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} 1 - \frac{min(P(t \in L_i), P(t \in L_j))}{max(P(t \in L_i), P(t \in L_j))} \tag{3.1}$$

[2] `dev.twitter.com`

[3] `www.let.rug.nl/vannoord/TextCat/`

Table 3.2: Hashtags Noisy Label

Positive	Negative	Neutral
#love	#fail	#news
#soexcited	#worstfeeling	#rainews
#excited	#sadtweet	#cnn
#bestfeeling	#worst	#bbc
#loveit	#sad	#job
#happy	#sosad	-
#sohappy	#hateit	-
#sogood	#hate	-
#bestfeelingever	#depressing	-
#happiness	#bored	-
#smile	#ihate	-
#amazing	-	-

where N is the number of the dataset labels, namely, $N = 3$ (i.e., positive, negative, and neutral) and $P(t \in L)$ is the likelihood that the term t belongs to the label class L. A zero value of *Salience* means that the term t appears uniformly in each dataset, thus it is a good candidate to be discarded. Finally, as for the Machine Learning algorithm, a Naïve Bayes classifier is trained on the training data, where each tweet is represented as a feature vector made up of the following groups of features:

- Bag-of-words: vectors of word unigram;
- Word polarities: using the LIWC[4] content analysis dictionary, we extracted features for positive, negative, and neutral words. Individual word polarities are inverted if the word follows a negation;
- Negations: we add the "NEG_" suffix to each word following a negation pattern (e.g., "not perfect" becomes 'perfect_NEG') according to the approach proposed in [37];
- Elongated words: we represent as a feature the presence of words with one character repeated more than two times, (e.g., "looove", "yesss");
- Part-of-speech tags: they provide a rough measure of the tweet content.

[4] liwc.net

3.4 SVO Recommendation Approach

3.4.1 User profiling

Several approaches to user recommendation are based on the definition of a similarity measure between two users u_i and u_j. Given the user u_i, the ranked list of suggested users corresponds to the set of users u_j that maximize the aforementioned measure. Content-based approaches define this measure by analyzing the user tweets. The set T of tweets $tweets(u)$ posted by the user u can be viewed as an extension of the *bag-of-word* model, where concepts are more semantically significant and less ambiguous than plain keywords. Instead of using complex semantic annotators, a concept is uniquely identified through *hashtags* contained in the tweet, namely, the metadata tags that are used in Twitter to indicate the context or the flow a tweet is associated with. Thus, we define the profile p of the user u as the set of weighted concepts:

$$p(u) = \{(c, \omega(u, c)) | c \in C_u\} \tag{3.2}$$

where $\omega(u, c)$ is the relevance of the concept c for the user u, and C_u is the set of concepts cited by the user u. The weighting function will be discussed in the following section.

The user profile representation is generated by monitoring the user activity, that is, all the tweets included in the observation period. Afterwards, given two users u_i and u_j, and their profiles $p(u_i)$ and $p(u_j)$, the similarity function is defined in terms of cosine similarity:

$$\begin{aligned} sim(u_i, u_j) &= sim(p(u_i), p(u_j)) = \\ &= \frac{\sum_{c \in C_{u_i} \cup C_{u_j}} \omega(u_i, c) \cdot \omega(u_j, c)}{\sqrt{\sum_{c \in C_{u_i}} \omega(u_i, c)^2} \cdot \sqrt{\sum_{c \in C_{u_j}} \omega(u_j, c)^2}} \end{aligned} \tag{3.3}$$

where C_{u_i} and C_{u_j} are the concepts in the profiles of users u_i and u_j, respectively.

3.4.2 SVO Weighting Function

The idea behind this work is that taking into account user attitudes towards his own interests can yield benefits in recommending friends to follow. Specifically, we consider (i) which is the sentiment expressed by the user for a given concept, (ii) how much he is interested in that concept, and (iii) how much he expresses objective comments on it.

In our model the first contribution $S(u, c)$, namely, the *sentiment* of the user u about a concept c, is obtained as follows:

$$S(u, c) = norm\left(\frac{Pos(u, c) - Neg(u, c)}{Pos(u, c) + Neg(u, c)}\right) \tag{3.4}$$

where $Pos(u, c)$ and $Neg(u, c)$ are the sums of the positive and negative tweets written by the user u regarding the concept c, respectively. Such values are calculated by means of our proposed Machine Learning algorithm (see Section 3.3) that classifies the tweets as positive, negative or neutral. A low value of $S(u, c)$ means that the user sentiments towards the concept c are negative, on the contrary a high value represents positive sentiments.

The *norm* function is used to normalize the output value within the [0, 1] range:

$$norm(x) = \frac{1}{1 + (k^{-x})} \tag{3.5}$$

where k = 10.

The second contribution is the *volume* $V(u, c)$, that is, how much a user u wrote about a specific concept c and is defined as follows:

$$V(u, c) = \frac{tweets(u, c)}{\sum_{i=1}^{N} tweets(u, c_i)} \tag{3.6}$$

where $tweets(u, c)$ is the number of tweets written by the user u about a specific concept c, and N is the total number of concepts dealt with by u.

The third contribution is the *objectivity* $O(u, c)$. With this term we denote how many tweets about a concept c are objective, namely, do not contain sentiments or opinions. This may be important because neutral tweets are tipically news, so quite significant for the similarity of user profiles but less relevant for the sentiment analysis.

$O(u, c)$ is defined as follows:

$$O(u, c) = \frac{Neutral(u, c)}{Pos(u, c) + Neg(u, c) + Neutral(u, c)} \tag{3.7}$$

where $Pos(u, c)$, $Neg(u, c)$ and $Neutral(u, c)$ are the sums of the positive, negative, neutral tweets written by the user u relative to the concept c, respectively.

Based on such contributions, we proposed a novel weighting function, we called *sentiment-volume-objectivity (SVO)* function, that takes into account all of them. It is defined as follows:

$$SVO(u, c) = \alpha S(u, c) + \beta V(u, c) + \gamma O(u, c) \tag{3.8}$$

where α, β, and γ are three constants $\in [0, 1]$, such that $\alpha + \beta + \gamma = 1$. The function $SVO(u, c) \in [0, 1]$ is the weighting function $\omega(u, c)$ that appears in the Equations 2 and 3.

The experimental evaluations (Section 3.5) shows the computation of the values of the parameters α, β, and γ that maximize the performance of the recommender.

3.5 Experimental Evaluation

3.5.1 Dataset

In order to evaluate the proposed model, we considered a case study rich of sentiments, such as the 2013 Italian political elections. Using the Twitter APIs we selected 31 hashtags for retrieving the Twitter streams about politician leaders and parties from Jan 25th to Feb 27th. Furthermore, because social networks are dynamic and fast-changing, we retrieved the hashtags that more often co-occur in the obtained tweets and added them to the initial hashtag set. This way, we took into account the trending topics that may be ignored in the initial query setup. The final dataset counted 1085000 tweets, meaning over 30000 tweets per day. The number of Twitter messages posted per user follows a power-law distribution. For the experimental evaluation we finally selected 1000 users that (i) posted at least 10 tweets in the observed period, and (ii) had more than 15 friends and followers already stored into the dataset.

3.5.2 Evaluation

The goal of our user recommender system is to suggest to a user someone to follow, with similar interests and opinions. In order to compare different profiling approaches and recommendation strategies, we need to understand when a user u_1 is relevant for a user u_2. In this work we suppose that u_1 is relevant for u_2 if a *following relationship* exists between them. This assumption has recently became a commonplace among social networks recommender systems [2, 96, 15] and is supported by the phenomenon of *homophily*, that is, the tendency of individuals with similar characteristics to associate with each other.

We performed a preliminary evaluation in order to assess the effectiveness of the proposed approach. For the sake of brevity, in this paper we only report the results of a comparative analysis of our approach with two traditional approaches that do not consider sentiment: (i) cosine similarity in a Vector Space Model (VSM) where vectors are weighted hashtags, and (ii) the function $S1$ proposed by Hannon*et al.* [84]. We used different metrics to express the evaluation results. *Success at Rank K* (S@K) provides the mean probability that a relevant user is located in the top K positions of the list of suggested users. *Mean Reciprocal Rank* (MRR) indicates the average position of a user in the recommended list. *Mean Average Precision* at cut-off K (MAP@K) is the average of the precision value for each of the top-K recommended users. Figure 3.1 shows the obtained evaluation results. As can be seen, our approach outperforms the other ones according to each evaluation metric. These findings confirm that sentiment is a valuable feature to be considered in order

to improve the user recommender systems. As a marginal note, the absolute values of the achieved results are high due to the characteristics of the built dataset, where the relations among users are significantly dense. Finally, we also analyzed the user recommender performance in terms of variations of the three parameters α, β, and γ (see equation 3.8). In order to determine the best values of those parameters, we implemented a *mini-batch gradient descent* algorithm. The best results, according to aforementioned metrics, was achieved running the evaluation with $\alpha = 0.3$, $\beta = 0.6$, and $\gamma = 0.1$. Based on the proposed model and the used dataset, these weights appear to highlight the contribution of the *volume* and the *sentiment* in comparison with the *objectivity*.

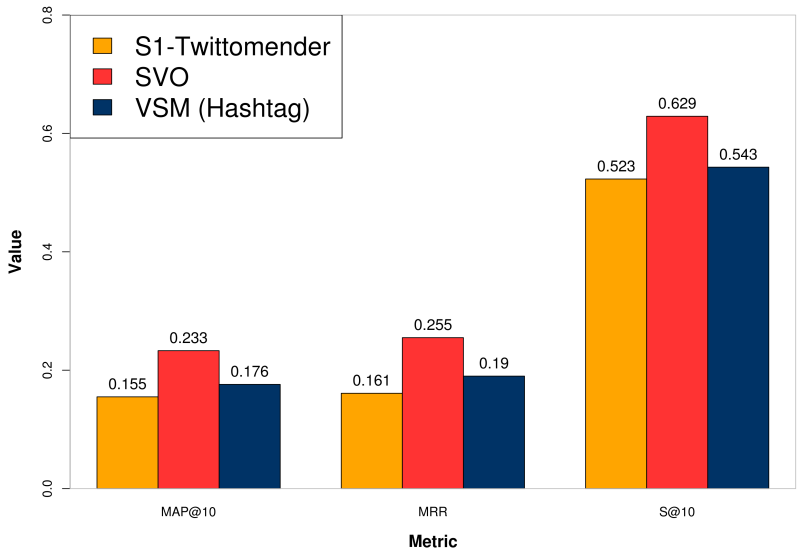


Figure 3.1: Comparative analysis among the proposed approach and two other state-of-the-art methods.

3.6 Summary

In this paper we have described a user recommender system for Twitter. Our work emphasizes the use of implicit sentiment analysis in order to improve the performance of the recommendation process. We have defined a novel weighting function that takes into account sentiment, volume, and objectivity related to the user interests. This technique allowed us to build more complete user profiles than traditional content-based approaches. Preliminary results show the benefits of our proposed model compared with some state-of-the-art methods.

As future work we are planning a deep sensitivity analysis to investigate whether social interactions, user preference and dataset characteristics shape parameters α, β, and γ. We will also include some improvements of the recommendation process taking into account other elements (e.g., named-entities, persons, products) and semantic representations of hashtags (e.g., [24]). A future study will also focus on the use of the implicit sentiment analysis using graph-based techniques and into collaborative filtering approaches.

Chapter 4

Exploiting Signals and Temporal Dynamics for a People-to-People RS

Apart from few notable exceptions, traditional user modeling approaches pay no attention to the temporal dynamics of changing interests, thereby ignoring important information that may be useful to better characterize a specific user. In this paper we describe a preliminary investigation of a novel user model, called *bag-of-signals*, that can also represent how user interests vary over time for creating more comprehensive user profiles. Such model involves the use of the wavelet transform, a signal processing technique that captures the frequency content of any signal, together with their precise location of occurrence in the time domain. As case study, we employ the advanced model to build a recommender system of new users to follow in social media, focusing on the case of Twitter. After evaluating the performance of the proposed model, we added another signal dimension that represent the sentiment of a user toward a specific topic, as presented in the previous Chapter 3. A comparative analysis, using real-user data, with some state-of-the-art techniques reveals the benefits of the proposed user modeling approach, and the addictional benefits of the *SVO* weighting schema.

4.1 Introduction

User interests and preferences constantly evolve as the ecosystem including users and objects of their interest is ever changing. Users may influence each other and their inclinations may alter over time. Therefore, tracking the temporal dynamics of user interests could help improve personalized systems. However, the so-called *monolithic* profile representations do not include the time patterns of changing interests. Indeed, their representation and analysis bring significant challenges. As possible solution we propose a novel user model, called *bag-of-signals*, which relies on a signal processing tool that seems tailor made to

carry out this task: the discrete wavelet transform (DWT). Thus, we employed our proposed user model in a user recommender on Twitter [1] and evaluated its performance in comparison with some classic approaches. After evaluating the performance of proposed approach, we added another signal dimension that represent the *SVO* of a user toward a specific topic. The Sentiment Analysis model and weighting schema was build as proposed in the previous Chapter 3. The research questions we pose are (i) can the consideration of temporal patterns of changing users interests really impact the characteristics and quality of user recommender? (ii) can Sentiment Analysis yield some benefits to the proposed temporal-based RS?

4.2 Related Work

In literature there exist not many studies that explicitly consider the temporal dynamics of changing user interests. Research on collaborative filtering shows the importance of their integration in recommendations algorithms [107]. Abel et al. report an in-depth analysis of topics discussed on Twitter and advance a user modeling framework for creating user profiles able to take into account their time patterns [5]. Hannon et al. propose a content-based approach that - like the present work - starts from a document-based representation of the user tweets [80]. They also extend it to include follower and following tweets, so enriching the user profile. Furthermore, almost the same authors have addressed the monolithic profile issue as well [83]. Instead of adopting a temporal profiling approach, however, they put forward a faceted profile structure that makes different types of interest more explicit. A first preliminary attempt of using the wavelet theory for recommendation tasks has been proposed in [22, 23]. The authors suggest a comparison among time habits in order to improve traditional collaborative approaches for movie and music recommendation. In addiction there are others version of the presented model which consider wavelet and signal that were published during this Ph.D. thesis [16, 29].

4.3 Bag-of-Signals User Model

Inspired by the work described in [140], we cast the problem of user representation into the problem of document representation. Such assumption allows us to borrow Information Retrieval (IR) techniques.

The context of user recommendation on Twitter requires some definitions. We define *pseudo-document* related to a user $u \in U$ the set of all tweets $x \in X$ posted by u in a given

[1] https://twitter.com

observation period ΔT

$$PseudoDoc(u, \Delta T) = \{x \in X \,|\, author(x) = u, time(x) \in \Delta T\} \tag{4.1}$$

with U set of all users and X set of all tweets. A natural extension of the well-known *bag-of-words* representation is the *bag-of-concepts* representation, where *concepts* instead of keywords are extracted from pseudo-documents. In this work, we consider the following types of concepts: (i) words or sentences prefixed with the symbol #, namely, *hashtags*, (ii) atomic elements in text classifiable into predefined categories (e.g., names of celebrities, places, events, companies), namely, *entities*, and (iii) subjects of tweets, namely, *topics*. In our approach, we employed *OpenCalais* [2] for entity extraction and *WikipediaMiner* [3] for topic detection and disambiguation. Then, we define *bag-of-concepts* user model of a user $u \in U$ the set of weighted concepts $c \in C_u$, with C_u set of all concepts extracted from all tweets posted by u in ΔT

$$P_{BoC}(u) = \{c \cdot w(u, c), \forall c \in C_u\} \tag{4.2}$$

where $w(u, c)$ is a function that gives the weight of the concept c for the user u. Now we have all the elements to define a new representation, which we call *bag-of-signals* to emphasize that the user model is made up of a set of signals, each of which is related to a different concept expressed by the user. Such representation is directly generated from the user activity, that is, all tweets posted by him during the observation period ΔT. In order to ensure the construction of the signals, it is necessary to extend the definition of pseudo-document to a more specific one: given a set of consecutive and same length intervals $\Delta t_j \in \Delta T$, with $j = [1, 2, \cdots, N]$, and

$$\Delta t_1 + \Delta t_2 + \cdots + \Delta t_N = \Delta T \tag{4.3}$$

we define *pseudo-fragment* related to a user $u \in U$ and an interval Δt_j, the set of all tweets x posted by u in Δt_j

$$PseudoFragment(u, \Delta t_j) = \{x \in X \,|\, author(x) = u,\ time(x) \in \Delta t_j\} \tag{4.4}$$

Then, we define a *signal* $s_{u,c}$ related to a user $u \in U$ and a concept $c \in C_u$ as follows:

$$s_{u,c} = [S_{u,c}(\Delta t_1), S_{u,c}(\Delta t_2), \cdots, S_{u,c}(\Delta t_N)] \tag{4.5}$$

The value of each signal component $S_{u,c}(\Delta t_j)$ is determined by a weighting function $\omega(u, c, \Delta t_j)$ where u is a user, c a concept, and Δt_j an interval.

[2] http://www.opencalais.com/

[3] http://wikipedia-miner.cms.waikato.ac.nz/

This function is used to give more importance to the concepts that appear more times in the user pseudo-document, but that are generally less frequent. For its definition we followed a traditional *tfidf* approach; the function $\omega(u, c, \Delta t_j)$ is therefore given by the product of two components as follows:

$$\omega(u, c, \Delta t_j) = cf_{u,c,\Delta t_j} \cdot ipf_{c,\Delta T} \tag{4.6}$$

The first component, which we call *concept-frequency* function, is defined as follows:

$$cf_{u,c,\Delta t_j} = \frac{|n_{c,\Delta t_j}|}{max_{c \in C_u, j \in [1,N]}\{n_{c,\Delta t_j}\}} \tag{4.7}$$

where $|n_{c,\Delta t_j}|$ is the frequency of the concept c in a specific pseudo-fragment, $max_{c \in C_u, j \in [1,N]}\{n_{c,\Delta t_j}\}$ is the frequency of the most frequent concept c within all pseudo-fragments related to the user u. The second component of the weighting function $\omega(u, c, \Delta t_j)$, named *inverse-period-frequency* function, is defined as follows:

$$ipf_{c,\Delta T} = log\left(\frac{|\Delta t_j \in \Delta T|}{|\Delta t_j : c \in \Delta t_j|}\right) \tag{4.8}$$

where $|\Delta t_j \in \Delta T|$ is the total number of pseudo-fragments in ΔT (i.e., N), $|\Delta t_j : c \in \Delta t_j|$ is the number of pseudo-fragments in ΔT wherein concept c appears. In the bag-of-concepts model, a user is represented through a set of *concepts* weighted according to their occurrences within the pseudo-document. In the proposed model, a user is represented by a set of *signals* related to several concepts extracted from tweets posted by u in the observation period. We define *bag-of-signals* user model of a user $u \in U$ the set of signals $s_{u,c}$ related to u, whose components $S_{u,c}(\Delta t_j)$ are determined by the weighting function $\omega(u, c, \Delta t_j)$

$$P_{BoS}(u) = \{s_{u,c}, \forall c \in C_u\} \tag{4.9}$$

Hence, the elementary units of the bag-of-signals representation are signals, each of which related to one of concepts $c \in C_u$.

These signals show strong discontinuities and sharp spikes. Signal processing provides an ideal tool for representing and analyzing such kind of signals: the discrete wavelet transform (DWT) [68]. Wavelets are mathematical functions that may be located both in time (space), as well as in scale (frequency), thus providing an accurate *time-scale map* of the signal. The wavelet-based analysis relies on the use of a prototype function, so-called *mother wavelet*, whose translated and scaled versions constitute the basis functions for the series expansion that ensures the representation of the original signal through coefficients. Operations involving signals can, therefore, be developed - in a more streamlined and efficient way - directly on corresponding wavelet coefficients. If the mother wavelet is properly

selected (in our approach we chose the Haar wavelet for its compact support), the wavelet transform allows for best capturing signal dynamics. Computation of the wavelet transform can be performed in a fast way (with computational cost $O(n)$, if n is the number of signal samples) by means of the *fast DWT* [120]. In our approach we consider the approximation $A_l(s)$ of the signal s at level *l-th*, which is defined by the set of approximation coefficients of the DWT limited to the level *l-th*

$$A_l(s) = \{a_{s,l}[j], \quad j = 1, ..., 2^l\} \tag{4.10}$$

Given two users u_1, u_2 and their respective bag-of-signals profiles $P_{BoS}(u_1)$, $P_{BoS}(u_2)$, several different wavelet-based similarity functions $sim(u_1, u_2)$ can be defined. In our previous work [17], we tested some of them obtaining the best results with

$$sim(u_1, u_2) = \frac{\sum_{c \in (C_{u_1} \cup C_{u_2})} \xi(s_{u_1,c}) \cdot \xi(s_{u_2,c}) \cdot \overline{z_l}(s_{u_1,c}, s_{u_2,c})}{\sqrt{\sum_{c \in C_{u_1}} \xi^2(s_{u_1,c})} \cdot \sqrt{\sum_{c \in C_{u_2}} \xi^2(s_{u_2,c})}} \tag{4.11}$$

where C_{u_1} and C_{u_2} are the sets of concepts related to the signals $s_{u_1,c} \in P_{BoS}(u_1)$ and $s_{u_2,c} \in P_{BoS}(u_2)$ respectively, the function $\xi(s)$ represents the energy of the signal s, and $\overline{z_l}(s_1, s_2)$ is a function that specifies how much the signals s_1 and s_2 have similar time use patterns. Given a discrete-time signal s, limited and with N real components $S(i)$, the *energy* $\xi(s)$ of the signal s is defined as follows:

$$\xi(s) = \sum_{i=0}^{N} S(i)^2 \tag{4.12}$$

The function $\overline{z_l}(s_1, s_2)$ in Equation 4.11 returns a value between 0 and 1, providing a measure of how much the concepts belonging to two different profiles have been used with similar time patterns. Such function is obtained as follows: given two signals s_1 and s_2 and their respective DWT approximations at level *l-th* $A_l(s_1)$ and $A_l(s_2)$ (see Eq. 4.10), we have

$$z_l(s_1, s_2) = \sum_{j=1}^{|2^l|} a_{s_1,l}[j] \cdot a_{s_2,l}[j] \tag{4.13}$$

whose normalized version $\overline{z_l}(s_1, s_2)$ that appears in Equation 4.11 is given by

$$\overline{z_l}(s_1, s_2) = \frac{z_l(s_1, s_2)}{\sqrt{z_l(s_1, s_1) \cdot z_l(s_2, s_2)}} \tag{4.14}$$

4.3.1 SVO Signal

In order to evaluate also in this work the effectiveness of sentiment analysis in a recommender systems, we integrated the *SVO* weighting model presented in Chapter 3 in the aforementioned bag of signal model. The weighting function is calculated as follows:

$$\omega(u, c, \Delta t_j) = \alpha S_{\Delta t}(u, c) + \beta V_{\Delta t}(u, c) + \gamma O_{\Delta t}(u, c) \tag{4.15}$$

where SVO values are the same expressed in Chapter 3.4.2 and the used similarity function refers to Equation 4.11.

4.4 Experimental Evaluation

Testing a user recommender on a social network like Twitter raises many challenges that it is not possible to discuss here for lack of space. In short, the basic idea we followed has been to exploit social relationships between users in order to establish if a user is really relevant for another one. We made a strong assumption: a user u_1 is relevant for a user u_2 if exists a *following relationship* between them. This hypothesis is supported by the phenomenon of *homophily* according to which two *similar* users have more probability to follow each other than two *not similar* users.

The dataset used for the experimental efforts has been obtained starting from the one proposed and employed in [2]. We enriched that dataset following those 20,000 users from October 2014 to January 2015. We selected only the 1500 users that posted at least ten tweet at month and at least 20 tweets in the whole observation period, obtaining more than 1 million tweets.

In the experimental tests we have used the *Success at Rank K (S@K)* and *Mean Reciprocal Rank (MRR)* as evaluation measures. *S@K* provides the mean probability that a relevant user is located in the top *K* positions of suggested users, *MRR* indicates the average position of a user in the recommended list.

To evaluate the effectiveness of the proposed approach we performed several preliminary tests, whose results can not all be reported for space reasons. Hereafter, we include the results of a comparative analysis (see Fig. 4.1 and Fig. 4.2) between the approach based on bag-of-signals model and two traditional methods that do not consider the time dimension: (i) cosine similarity in a *Vector Space Model (VSM)* where vectors are weighted concepts, and (ii) the *function S1* proposed in [80], which is based on a vector user representation.

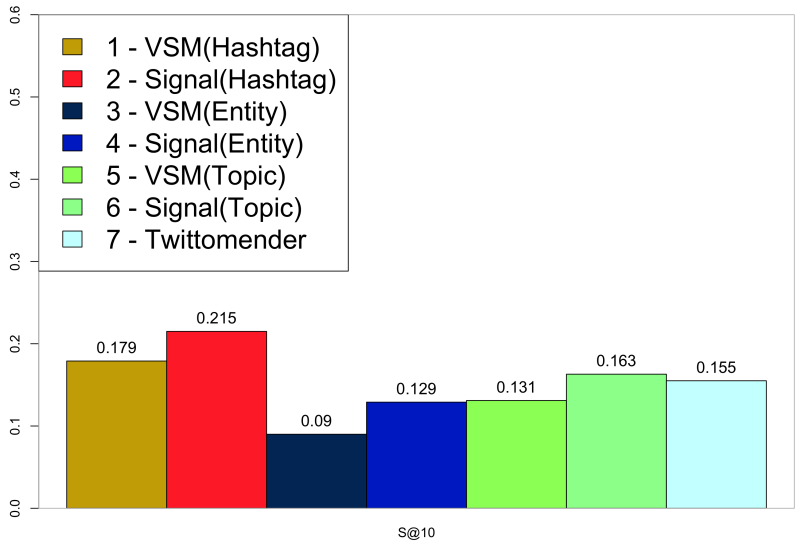


Figure 4.1: Results of a comparative analysis among our approach and two classical techniques advanced in literature, that is, *Vector Space Model (VSM)* and *function S1* (see [80]), in terms of *S@10*

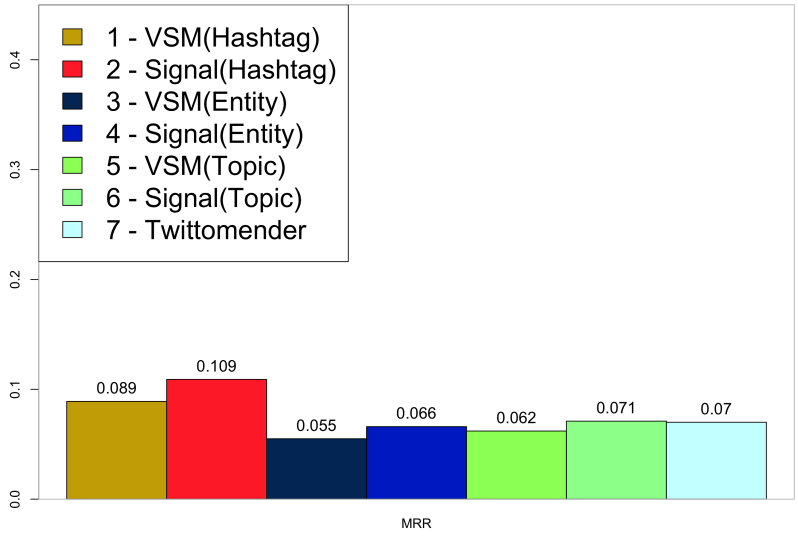


Figure 4.2: Results of a comparative analysis among our approach and two classical techniques advanced in literature, that is, *Vector Space Model (VSM)* and *function S1* (see [80]), in terms of *MRR*

Firstly, it can be noted that our approach based on hashtags performed significantly better than the same approach based on entities and concepts. This might seem an unexpected result, because entities and concepts should be more semantically significant than simple hashtags. Moreover, in principle entities and concepts should appear in users activities more often than hashtags. Indeed, while some users do not use hashtags, most of them report entities in their posts, and topics should be extracted from any tweet. However, analyzing the models constructed from the two different types of concepts, we found out that entity-based profiles were composed of 63 signals on average, concept-based profiles of 85 signals, while hashtag-based profiles were composed of 223 signals.

Hence, our theory is that the smaller amount of information in case of entities and concepts, resulted in worse results than those obtained by extracting hashtags. This result also shows how the Twitter hashtag mechanism is well-established and widespread, and then it can be usefully exploited for user profiling purposes. Furthermore, it can be observed that our approach performs definitely better than baselines. These findings confirm that harnessing the time dimension can guarantee better results in user profiling.

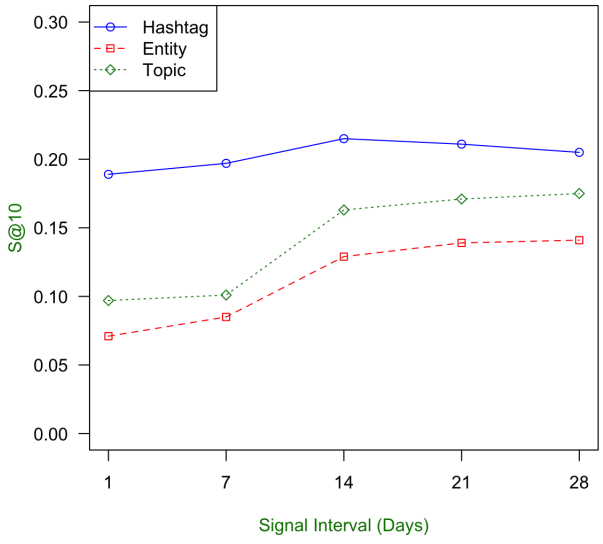


Figure 4.3: Performance of the proposed approach in terms of *S@10* for different values of the signal interval.

We also analyzed the performance of the user recommender varying signal intervals from 1 day to 28 days. This important parameter of the model is the length of the signal interval, namely, the number of days for each sample whereby the signals have been generated. The results illustrated in Figure 4.3 and Figure 4.4 show different behaviours for

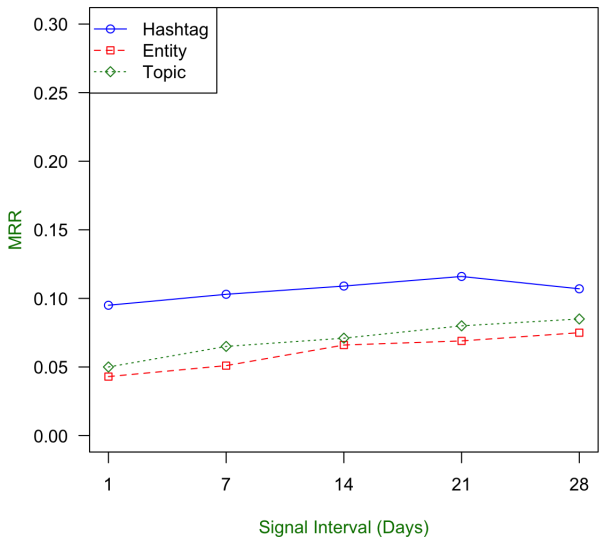


Figure 4.4: Performance of the proposed approach in terms of *MRR* for different values of the signal interval.

hashtags, entity and topic profiling. Hashtags maintain flat results thanks to the amount of information extracted even with 1 or 7 days of signal interval. On the other hand increasing results for topic and entity profiling, show that for shorter signal intervals there are not enough semantic information to profile users as well.

4.4.1 SVO Comparison

To compare Signal-SVO with our previous Bag of Signal approach we continue collecting the presented dataset from January 2016 to December 2016. This datasets counts almost 3 million tweets, and we selected 800 users to build a test subset. The signal model considered for the comparison is the one explained in section 4.3 using hashtag-based user profile. The evaluation metrics we considered for this test was the S@10 metric, with a signal intervals of 28 days.

As can be noted in Figure 4.5 the contributions of sentiment, volume, and objectivity seem to slightly improve the recommender precision. The lower value of S@10 is due to the selected subset of 800 users instead of taking all of 1,500 users considered in the previous tests. Also note that during the writing of this dissertation, further evaluation was started but cannot be reported yet in this thesis.

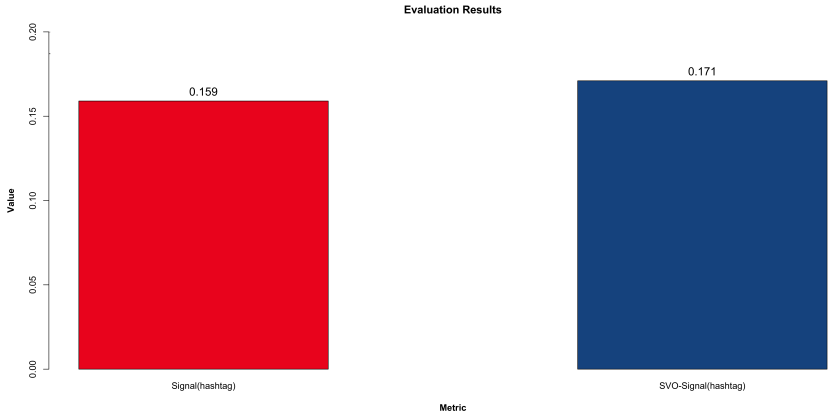


Figure 4.5: Comparison between Signal-based approach and Sentiment-Signal approach.

4.5 Summary

In this paper we have described a user recommender system on Twitter. Such a system is based on a novel user model, termed *bag-of-signal*, which makes use of signal processing techniques to represent not only the number of occurrences of the informative entities (concepts), but also the related time use patterns. The bag-of-signal user model involves modeling the user interests through a set of signals and the adoption of similarity functions suitably defined. Specifically, for the signal analysis and representation we employ the wavelet mathematical tool for its main characteristic of time-frequency localization. From a practical point of view, the discrete wavelet transform allows us to effectively analyze the sampled signals with a different time window. Such model has been adopted for user recommendation on Twitter and the obtained findings allow us to answer the research question positively. In addiction, as also found in the previous work presented in Chapter 3, the contribution of *SVO* and specifically of the Sentiment Analysis slightly improve the precision of the recommendation.

The experimental evaluation has also shown the inadequacy of the tools used for extracting certain types of concepts, such as the named entities, from a noisy source like the stream of Twitter. This fact suggests some possible developments of the proposed approach. Specifically, we could obtain a greater number of named entities by using new tools of information extraction, thus producing semantically more meaningful user profiles. A further possible future development could involve the extraction and classification of other types of concepts, such as topics. As for the experimental tests, a future work will consider the employment of the bag-of-signals user model for different tasks, such as personalized news recommendation.

Chapter 5

Leveraging Community Detection Techniques for User RS

Among the various recommender systems proposed in the literature, there is an increase in relevance and number of those that suggest users of possible interest to the target user. In this article, we propose a new algorithm for realizing user recommenders, named *SCORES (Sentiment COmmunities REcommender System)*. This algorithm relies on the identification of sentiment communities in which, for each topic cited by the user, we consider not only the relative sentiment, but also the volume and the objectivity of contents generated by him. The graph related to each topic is obtained by considering the Tanimoto similarity between users. Clustering based on the modularity optimization allows us to detect the latent communities. The recommendation process occurs by suggesting to the target user the most similar K users based on the tie strength measure. A comparative analysis between *SCORES* and some state-of-the-art approaches shows the benefits in term of performance. The experimental evaluation performed on real-world datasets gathered from Twitter also reveals how such benefits can be further improved by choosing the most appropriate tie strength measure and tuning the weights of the contributions of sentiment, volume, and objectivity, according to the different category of topics dealt with by the user.

5.1 Introduction

With the proliferation of user-generated contents on social media such as reviews, discussion forums, blogs, and tweets, detecting sentiments and opinions from the Web is becoming an increasingly widespread form of data interpretation. In particular, sentiment analysis aims to understand subjective information, such as opinions, points of view and feelings expressed by users in the content they generate. Sentiment analysis in social media enables companies and organizations to monitor their business reputation, identify public opinions

regarding products and services of themselves and their competitors, gain insights about possible emerging trends and changes in market opinions, etc.

In this article, we propose exploiting sentiment analysis for a new task: the identification of latent communities and their subsequent use in recommending similar users to the target user, that is, the user we want to suggest someone to follow. The research questions underlying our work are, therefore, the following:

1. Can the consideration of sentiment bring benefits for recommending users to follow?
2. If so, what is the best approach to do this?
3. Are there differences depending on the category of topics dealt with by the user?

Hereafter we see how the process of user recommendation can be indeed improved through the use of sentiment. In particular, we show how this can be done by means of the identification of latent communities of users that, instead of considering social relationships, takes into account not only the user's sentiment towards the dealt with topics, but also the volume and the objectivity which he expresses in generated contents. Based on such contributions, we define a *sentiment-volume-objectivity (SVO)* function. Hence, our method relies on (i) the construction of graphs, one for each topic cited by the user, (ii) the detection of the SVO-based latent communities through clustering techniques based on modularity optimization, (iii) the use of different measures of tie strength to adopt in the computation of the global similarity between users.

The experimental tests were performed on different real-world datasets, obtained by monitoring the traffic produced by users on Twitter [1]. Such data enabled us to realize a comparative analysis of our system, called *SCORES (Sentiment COmmunities REcommender System)*, with similar approaches proposed in the literature. Test results allowed us to make also some remarks about the category of topics dealt with by the users. More specifically, we observed how the performance of SCORES could be improved by tuning the weights that define the different contributions of sentiment, volume, and objectivity, based on the nature of topics on which the similarity between users is computed.

The paper is structured as follows. Section 5.2 reviews some related works. Sentiment analysis is described in Section 5.3, community detection in Section 5.4, whilst the user recommendation process is presented in Section 5.5. Section 5.6 reports the results of the experimental efforts, including a comparative analysis between the proposed approach and some traditional techniques. Section 5.7 concludes and outlines some possible future works.

[1] `twitter.com`

5.2 Related Work

With the exponential advancement of social media and networking sites, sentiment analysis is increasingly being applied for the task of social network analysis. In particular, data extracted from Twitter are the subject of several contributions in the literature, which aim to analyze sentiment for several purposes such as prediction in political elections [167], event identification [165], and location recommendation [186]. As far as we are aware, however, there have been few attempts to consider user attitudes in micro-posts for community detection or user recommendation. In [183] the authors view the problem of community sentiment discovery as a semidefinite programming (SDP) problem and as an optimization problem, and solve both of them through a SDP-based rounding method. Nguyen *et al.* [132] address the problem of clustering blog communities into groups, called *hyper-communities*, based on user sentiments, and propose a non-parametric clustering algorithm for its solution. Yuan *et al.* [188] provide an interesting study on how to make use of sentiment towards topics of common interest for link prediction between users. They put forward different techniques to assess how the *sentiment homophily* (i.e., the tendency of people to express similar levels of sentiment to that expressed by their friends, in comparison with the overall average [26, 164]) can help improve the prediction of the likelihood of two users mutually mention or follow each other. User recommendation approaches that ignore sentiment opinions have been proposed by Freyne *et al.* [55] and Chen *et al.* [33] exploring different recommendations strategies. Guy *et al.* [76] propose a people recommendation engine within an enterprise social network site scenario. They aggregate several different sources to derive factors that might influence the similarity measure. Twittomender [4, 81, 15] lets users find pertinent profiles on Twitter exploiting different strategies, both content-based and collaborative ones. Arru *et al.* [15] propose a signal-based representation of user interests in order to draw similarities among people.

5.3 Sentiment Analysis

Sentiment analysis or opinion mining is formally defined as the computational study of sentiments and opinions about an entity expressed in a text. According to Liu [116], the entities are classified into five categories: *product, person, brand, event, concept*. Particularly, in this work we assume the *concept* as the sentiment analysis target entity. Sentiment analysis is a difficult task, hence - before the setup of the algorithm - some assumptions are needed. There are multiple granularity levels of sentiment analysis, as explained in [9]:

feature-level, entity-level, sentence-level, document-level. In this work we consider sentiment analysis at sentence-level. More specifically, in the Twitter domain we assume that a sentence matches the whole tweet. Moreover, we assume that each sentence contains only one opinion related to the entity.

The goal of our sentiment analysis system is to obtain an output value that represents how much positive, negative or neutral is the sentiment expressed in a tweet. For this reason, we implemented a supervised machine learning algorithm based on a Naïve Bayes classifier. With a view to training our algorithm, we needed a dataset with labeled tweets. However, due to the lack of a Twitter public dataset, we decided to follow an alternative approach. Instead of manually building a labeled dataset, Bhayani *et al.* [65] propose to employ a noisy dataset of positive, negative, and neutral tweets.

The Twitter APIs[2] have been used to retrieve a set of tweets containing the aforementioned features. The final training dataset counts 150000 tweets divided in 50000 tweets for each class. Because the experimental evaluations are conducted on events in Italian and English, the TextCat language recognizer[3] is employed to limit the set to those languages. In order to increase the classifier precision and reduce the presence of noise, we further performed feature selections such as salience and entropy [137]. Finally, as for the machine learning algorithm, a Naïve Bayes classifier is trained on the training data, where each tweet is represented as a feature vector made up of the following groups of features:

- Bag-of-words: vectors of word unigram;
- Word polarities: using the LIWC[4] content analysis dictionary, we extracted features for positive, negative, and neutral words. Individual word polarities are inverted if the word follows a negation;
- Negations: we add the "NEG_" suffix to each word following a negation pattern (e.g., "not perfect" becomes "perfect_NEG_") according to the approach proposed in [37];
- Elongated words: we represent as a feature the presence of words with one character repeated more than two times (e.g., "looov", "yess");
- Part-of-speech tags: we provide a rough measure of the tweet content. We used the Stanford POS Tagger[5] for English tweets and Morph-it! Tagger[6] for Italian tweets.

[2] dev.twitter.com
[3] www.let.rug.nl/vannoord/TextCat/
[4] liwc.net
[5] http://nlp.stanford.edu/software/tagger.shtml
[6] http://sslmitdev-online.sslmit.unibo.it/linguistics/morph-it.php

5.4 Community Detection

The idea behind this work is that taking into account user attitudes towards his own interests can yield benefits in recommending friends to follow. Specifically, we consider (i) which is the sentiment expressed by the user for a given concept, (ii) how much he is interested in that concept, and (iii) how much he expresses objective comments on it. For *concept* we mean any entity hashtag extracted from a tweet that can somehow characterize it.

In our model the first contribution is the *sentiment* $S(u, c)$, which represents a feeling or opinion about a concept c expressed by the user u, with $u \in U$ (set of all users) and $c \in C_u$ (set of all concepts expressed by the user u), and is obtained as follows:

$$S(u, c) = norm\left(\frac{Pos(u, c) - Neg(u, c)}{Pos(u, c) + Neg(u, c)}\right) \tag{5.1}$$

where $Pos(u, c)$ and $Neg(u, c)$ are the sums of the positive and negative tweets written by the user u regarding the concept c, respectively. Such values are calculated by means of a supervised machine learning algorithm based on a Naïve Bayes classifier, introduced in Section 5.3. The *norm* function is used to normalize the output value within the [0, 1] range and its expression is the following:

$$norm(x) = \frac{1}{1 + (k^{-x})} \tag{5.2}$$

where k = 10. The second contribution is the *volume* $V(u, c)$, that is, how much a user u wrote about a specific concept c and is defined as follows:

$$V(u, c) = \frac{tweets(u, c)}{\sum_{i=1}^{C_u} tweets(u, c_i)} \tag{5.3}$$

where $tweets(u, c)$ is the number of tweets written by the user u about a specific concept c, and C_u is the total number of concepts dealt with by u. The third contribution is the *objectivity* $O(u, c)$, which expresses how many tweets about a concept c do not contain sentiments. $O(u, c)$ is defined as follows:

$$O(u, c) = \frac{Neutral(u, c)}{Pos(u, c) + Neg(u, c) + Neutral(u, c)} \tag{5.4}$$

where $Pos(u, c)$, $Neg(u, c)$ and $Neutral(u, c)$ are the sums of the positive, negative and neutral tweets written by the user u relative to the concept c, respectively. Based on such contributions, we define a *sentiment-volume-objectivity (SVO)* vector, which takes into account all of them. If we consider a user $u \in U$ and a concept $c \in C_u$, it is defined as follows:

$$\vec{SVO}(u, c) = [\alpha S(u, c), \beta V(u, c), \gamma O(u, c)] \tag{5.5}$$

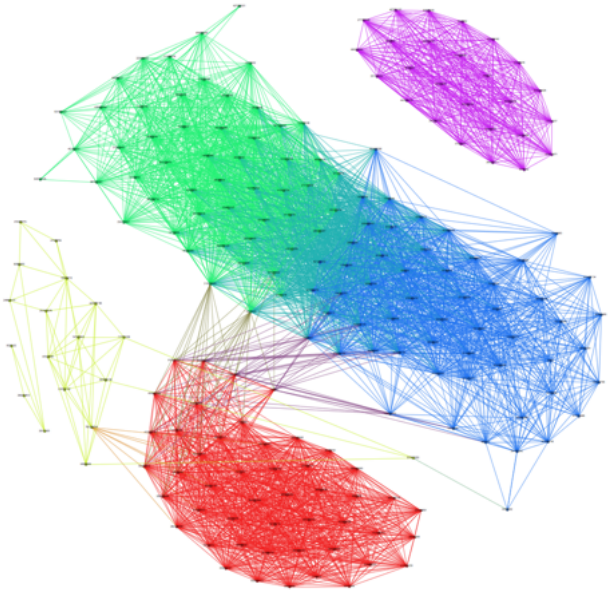


Figure 5.1: Communities identification in a specific topic graph.

where α, β, and γ are three constants in the [0, 1] interval, such that $\alpha+\beta+\gamma = 1$. In order to determine the optimal values of those parameters, we implemented a *mini-batch gradient descent* algorithm. In Section 5.6 we will see how such values depend on the category of topics mentioned by the user. For each concept c we compute the Tanimoto similarity [163] between users u and $v \in U$ as follows:

$$sim(u, v, c) = \frac{S\vec{VO}(u, c) \cdot S\vec{VO}(v, c)}{\| S\vec{VO}(u, c) \|^2 + \| S\vec{VO}(v, c) \|^2 - S\vec{VO}(u, c) \cdot S\vec{VO}(v, c)} \tag{5.6}$$

The similarity value lies in between [0, 1].

Once the similarities between users are computed, for each concept c we build a graph $G_c(V, E)$, where V represents the set of users, E the set of edges between them. We consider the similarity value as an edge between them, only if the similarity value between two users exceeds a threshold value Θ. Also the optimal value for Θ was determined through a gradient descent algorithm that maximizes the recommender precision. Afterwards we implemented a clustering algorithm based on modularity optimization that allows us to detect the latent communities for the considered concept c. In Figure 5.1 is showed the community identification, one for each color, for a specific concept c. This algorithm tends to iteratively optimize the modularity value. The modularity value has the following expression:

$$Q = \frac{1}{2m} \sum_{u,v} [A_{uv} - \frac{k_u k_v}{2m}] \cdot \delta(g_u, g_v) \tag{5.7}$$

where A_{uv} represents the weight of the edge between u and v, $k_u = \sum_v A_{uv}$ is the sum of the weights of the edges linked to the user u, g_u is the community to which user u is assigned,

$m = \frac{1}{2} \sum_{uv} A_{uv}$, and δ-function $\delta(s, t)$ is 1 if $s = t$ and 0 otherwise.

5.5 User Recommendation

Once identified the communities for all concepts mentioned by the target user u, the user recommender system works as follows. For every user v in the dataset, for each mentioned concept c we verify if it was also mentioned by the user u. In the positive case, we consider the related graph and calculate the measure of tie strength between u and v to obtain the recommendation score.

The notion of *tie strength* in social networks was introduced in [67]. Since then a lot of tie strength measures have been proposed in the literature. Hereafter we introduce those measures we employed in our recommender system.

Given a graph $G_c(V, E)$, we define *neighbor* of its node u a node with a direct link to u (i.e., a node with a path distance equal to one), and denote by $\Gamma(u)$ the set of its neighbors.

- Graph Distance. This measure is defined as follows:

$$TS(u, v) = length(path(u, v)) \tag{5.8}$$

  that is, the number of hops of the shortest path between u and v.

- Common Neighbors. This measure is equal to the shared neighbors between u and v.

$$TS(u, v) = |\Gamma(u) \cap \Gamma(v)| \tag{5.9}$$

- Jaccard Index. This measure normalizes the common neighbors with the total neighbors of u and v.

$$TS(u, v) = \frac{|\Gamma(u) \cap \Gamma(v)|}{|\Gamma(u) \cup \Gamma(v)|} \tag{5.10}$$

- Adamic-Adar. This measure, which was introduced by Adamic and Adar [7], is defined as follows:

$$TS(u, v) = \sum_{N \in \Gamma(u) \cap \Gamma(v)} \frac{1}{log|N|} \tag{5.11}$$

  where $\Gamma(u)$ and $\Gamma(v)$ are the neighborhoods of u and v respectively, and N is the number of nodes belonging to both of them.

- Weighted Adamic-Adar. This is a modified version of the previous Adamic-Adar measure, where we consider, as a weight, the average link weight between two users, namely, the SVO similarity:

$$TS(u,v) = \sum_{N \in \Gamma(u) \cap \Gamma(v)} \frac{1}{log|N|} \cdot Avg(\frac{1}{sim(\Gamma(u) \cap \Gamma(v))}) \tag{5.12}$$

where $sim(\Gamma(u) \cap \Gamma(v))$ is given by Equation 5.6, that is, the average edge weight of the common neighbors between u and v.

- Preferential Attachment. This measure represents the number of neighbors of u multiplied by the number of neighbors of v.

$$TS(u,v) = |\Gamma(u)| \cdot |\Gamma(v)| \tag{5.13}$$

In this case we do not take the community structure into account, but emphasize well-connected nodes over less connected ones.

- Katz. This measure sums over the entire collection of paths, each one exponentially damped by its length to emphasize short paths:

$$TS(u,v) = \sum_{l=1}^{\infty} \tau^l \cdot |paths_{u,v}^{<l>}| \tag{5.14}$$

where $|paths_{u,v}^{<l>}|$ is the set of all length-l paths from u to v. A very small value of τ yields a tie strength much like common neighbors, since paths of length three or more contribute very little to the summation.

- SimRank. This measure represents the similarity between two nodes u and v by recursively computing the similarity of their neighbors. It is parametrized by a constant $0 \leq \phi \leq 1$ as follows:

$$TS^{\phi}(u,v) = \begin{cases} 1 & \text{if } u = v \\ \phi \cdot \frac{\sum_{a \in \Gamma(u)} \sum_{b \in \Gamma(v)} TS(a,b))}{|\Gamma(u) \cdot \Gamma(v)|} & \text{otherwise} \end{cases} \tag{5.15}$$

- Random Walk with Restarts Algorithm. Starting at a node u, the algorithm faces two choices at each step: either moving to a randomly chosen neighbor with probability $1 - \mu$, or jumping back to the starting node with probability μ. The tie strength between u and v is the probability that the last node of the process is v.

To calculate the total score between two users u and v, we consider the sum of tie strength contributions for concepts mentioned by both of them:

$$Score(u, v) = \omega \cdot \sum_{c \in C_u \cap C_v} TS_c(u, v) \tag{5.16}$$

where

$$\omega = \frac{|C_u \cap C_v|}{|C_u \cup C_v|} \tag{5.17}$$

is the ratio between the number of concepts shared by u and v and all the concepts cited by u and v. In this way the contribution of users sharing more concepts with the target user is greater than others.

We evaluate the total score between the target user u and all the users v in the dataset, and suggest to him a ranked list of the most K relevant users based on such value. The pseudocode of the total process of SCORES is shown in Algorithm 1. We suppose that user u is the user we want to suggest someone relevant to follow.

5.6 Experimental Evaluation

5.6.1 Datasets

In order to comprehensively evaluate SCORES, we considered three datasets obtained from Twitter. Those datasets were gathered using the Twitter APIs searching for specific hashtags.

- **Dataset 1**

  $Dataset_1$ was obtained in 2013 during the Italian political elections. We retrieved the Twitter streams about politician leaders and Italian parties from Jan 25th to Feb 27th. Furthermore, because social networks are dynamic and fast-changing, we retrieved the hashtags that more often co-occur in the obtained tweets and added them to the initial hashtag set. This way, we took into account the trending topics that might have been ignored in the initial query setup. The final dataset counts 1,085,121 tweets written in Italian language and 70,977 unique users.

- **Dataset 2**

  $Dataset_2$ was obtained through the APIs searching for hashtags and keywords representing the most important mobile tech companies such as Samsung, Apple, Nokia, Huawei, LG, Motorola, and Blackberry. The dataset was gathered from Sep 2014 to Feb 2015 taking into account only Italian tweets, and counts 3,511,455 tweets from 181,000 unique users.

Algorithm 1: SCORES algorithm

input : $u \in U$ target user
$v \in U$ generic user in the dataset
$c \in C_u$ concept cited by u
Θ similarity threshold value
output: top-K recommended users to follow

```
/* Community detection                                    */;
foreach concept c do
    /* Sentiment, volume, and objectivity analysis        */;
    SVO(u,c) ⇐ S(u,c), V(u,c), O(u,c);
    foreach user v do
        SVO(v,c) ⇐ S(v,c), V(v,c), O(v,c);
        /* Calculate Tanimoto similarity (Equation 5.6)   */;
        sim(u,v,c) ⇐ SVO(u,c), SVO(u,c);
        /* Comparison with Θ                              */;
        if (sim(u,v,c) ≥ Θ) then
            /* Insert edge e between u and v into graph G_c */;
            G_c(V,E) ⇐ insert(e = (u,v)) ;
            /* Set weight of edge e                       */;
            e = (u,v) ⇐ sim(u,v,c);
    /* Identify communities for concept c graph           */;
    ClusteringAlgorithm(G_c(V,E));
/* Similarity computation between u and all users v       */;
foreach user v do
    foreach graph G_c(V,E) do
        Score(u,v) ← ω · Σ_{c∈C_u∩C_v} TS_c(u,v);
/* Select K users with the highest Score(u,v)             */;
return v_1, v_2, ..., v_k
```

- **Dataset 3**

  To facilitate the reproducibility of our approach we gathered the $Dataset_3$ with English tweets. This dataset was built searching terms that match the automotive landscape. Thus we searched terms such as Audi, BMW, Ferrari, Jaguar, Mercedes, Toyota, and Porsche. The collection set, retrieved from Dec 2014 to Feb 2015, counts 2,915,131 tweets from 110,350 unique users.

5.6.2 Datasets Analysis

As a first analysis on the obtained data, we wanted to investigate how sentiments can shape the creation of a relationship between users among different concepts. In order to do that, we analyzed the probability of two connected users sharing the same sentiment toward a specific concept, compared with two unconnected users. Two users are connected if they follow each other in the actual social graph. Furthermore, we assume that two users share the same sentiment if they write the equivalent number of positive or negative comments for a specific concept. Starting from the obtained datasets, for each concept (e.g. Audi, Bmw), we selected (i) 500 pairs of connected users and 500 pairs of unconnected users, and (ii) users that post at least five tweets for each concept.

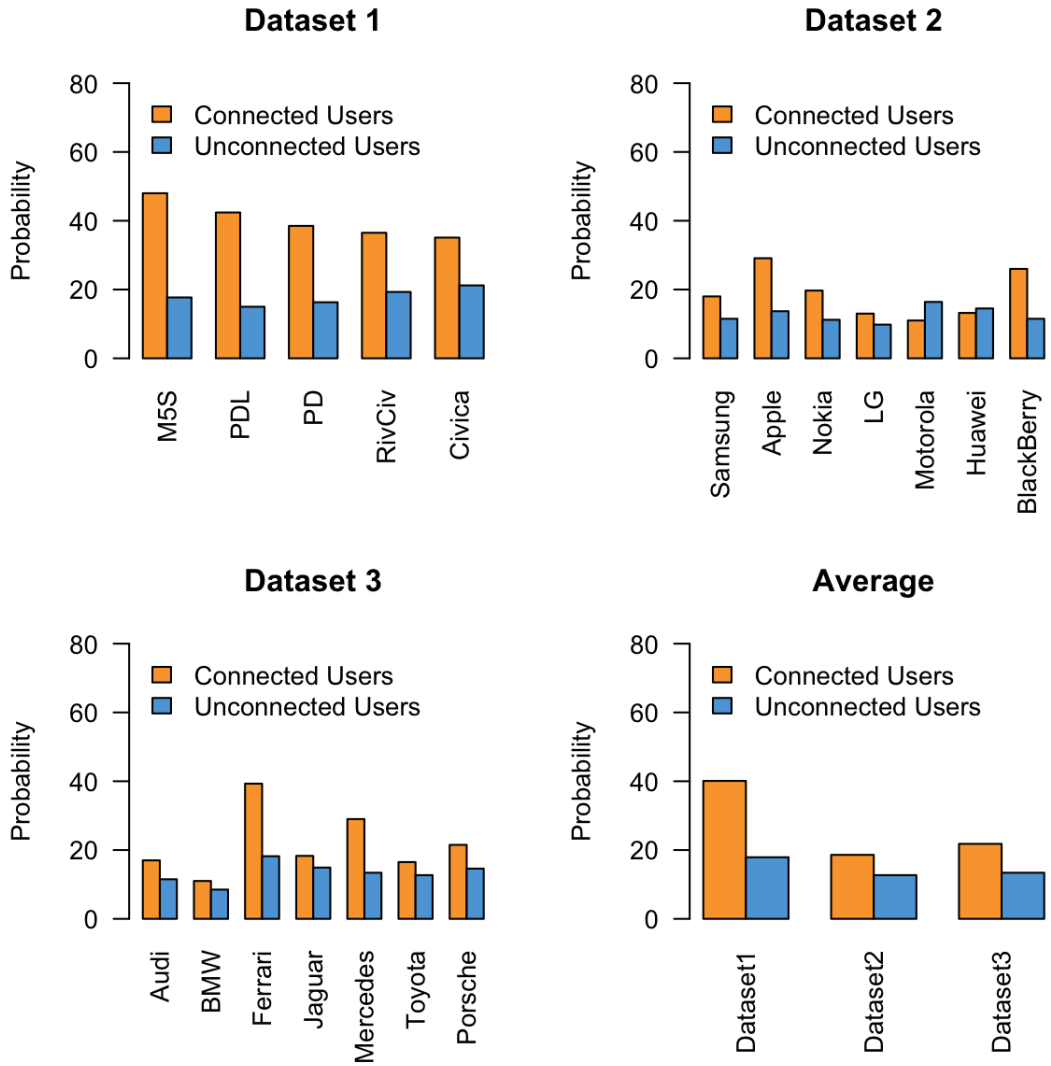


Figure 5.2: Probability of two users sharing a sentiment towards different concepts and datasets.

Figure 5.2 shows the obtained results. Regarding $Dataset_1$, it is interesting note how over 40% of connected users share the same sentiment toward a specific concept and, conversely, less than 18% of unconnected users share the same sentiment. This result was expected for $Dataset_1$ due to its political nature. Indeed, users sharing the same political sentiments (e.g., support the same political party) tend to establish a social relationship and belong to the same sentiment communities. As regards $Dataset_2$ and $Dataset_3$, the difference between the average probability of connected and unconnected users is less evident. The probabilities are only higher for Ferrari and Apple concepts, likely because also in this case users belong to sentiment communities that absolutely appreciate or not these brands. However, the average values of $Dataset_2$ and $Dataset_3$ indicate that in such cases sharing

sentiments is not a unique and decisive factor for the social relationships between users. Those findings are interesting because help understand that in some cases users tend to follow each other even if not sharing the same sentiment, but for example having opposite sentiments or different interests. This motivated our choice of testing the system on different datasets and various combinations of weights in the SVO function.

5.6.3 Results

The goal of our recommender is to suggest to a target user someone to follow. To compare different profiling approaches and recommendation strategies, we need to determine when a user u is indeed relevant to another user v. We suppose that u is relevant to v if a *following relationship* exists between them. This assumption has already been proposed in literature [4, 81, 15] and is supported by the phenomenon of *homophily*, that is, the tendency of individuals with similar characteristics to associate with each other. In order to evaluate our system we selected, for each dataset, 1000 users that (i) posted at least 50 tweets in the observed period, and (ii) had more than 30 friends and followers. We used the *Success at Rank K (S@K)* metric, which provides the mean probability that a relevant user is located in the top K positions of the list of suggested users. Table 5.1 shows the performance of our recommendation algorithm for different tie strength measures. Interestingly, our experimental evaluation enabled us to notice strong correlations among communities related to a specific dataset. The first analysis indicates that the best measures among all datasets are Weighted Adamic-Adar and Katz, but these results change while varying the dataset and, therefore, the concepts. Katz measure works best for topics about politics ($Dataset_1$) where the strong ties are very important. Indeed, if we use Katz tie strength, we suggest the most similar SVO users and, therefore, the nearest users within the same SVO community. On the contrary, in the other two datasets Weighted Adamic-Adar resulted the best tie strength measure, which is inversely proportional to the number of common neighbors and the SVO similarity. In this case, we are indeed suggesting the weak ties, that is, users that belong to different SVO communities with low similarity. These findings highlight that in less opinion-oriented topic such as technology ($Dataset_2$) and automotive ($Dataset_3$), recommending users belonging to a different SVO communities might be more useful. We plan to further investigate this issue in order to fully understand the real nature of those interactions and exploit such knowledge in the recommendation process.

In Table 5.2 we report the results of a comparative analysis of our system with some state-of-the-art functions. More precisely, we considered the following functions:

Table 5.1: Performance in terms of S@10 metric for different tie strength measures and different datasets (*τ = 0.2; **ϕ = 0.8).

Tie Strength	$Dataset_1$	$Dataset_2$	$Dataset_3$	Average
Graph Distance	0.155	0.151	0.159	0.155
Common Neighbors	0.202	0.172	0.169	0.181
Jaccard Index	0.178	0.162	0.167	0.169
Adamic-Adar	0.177	0.195	0.185	0.186
Weighted Adamic-Adar	0.179	**0.215**	**0.205**	**0.200**
Preferential Attachment	0.152	0.158	0.161	0.157
Katz*	**0.218**	0.191	0.189	**0.199**
SimRank**	0.165	0.156	0.159	0.160
Random Walk	0.175	0.161	0.165	0.167

Table 5.2: A comparison among different state-of-the-art techniques. The values of Θ similarity threshold are 0.821 for $Dataset_1$, 0.630 for $Dataset_2$, and 0.711 for $Dataset_3$.

Recommender System	$Dataset_1$	$Dataset_2$	$Dataset_3$
SCORES	**0.218**	**0.215**	**0.205**
S1-Twittomender	0.130	0.118	0.115
S7-Twittomender	0.172	0.163	0.161
VSM (Hashtag)	0.127	0.099	0.105
FoF	0.165	0.155	0.159

1. a content-based function, called *S1-Twittomender* [81], where users are profiled through the content of their tweets;

2. a collaborative filtering function, *S7-Twittomender* [81], where users are represented through a combination of followers and followees;

3. a *VSM (Hashtag)* function representing cosine similarity in a vector space model, where vectors are weighted hashtags;

4. a *Friend-of-Friend (FOF)* function proposed in [33], which leverages the social network information only, that is, user relationships (followers and followees).

As can be seen, our approach outperforms the other ones. These results confirm the potential of sentiment as a valuable feature for improving user recommender systems.

Finally, we also analyzed the user recommender performance in terms of variations of the three parameters α, β, and γ (see Equation 5.5). In order to determine the best values of those parameters, we implemented a *mini-batch gradient descent* algorithm and found the following values:

- $Dataset_1$: $\alpha_1 = 0.45, \beta_1 = 0.45$, and $\gamma_1 = 0.10$
- $Dataset_2$: $\alpha_2 = 0.25, \beta_2 = 0.50$, and $\gamma_2 = 0.25$
- $Dataset_3$: $\alpha_3 = 0.28, \beta_3 = 0.52$, and $\gamma_3 = 0.20$

Based on the proposed model and the used datasets, these weights appear to highlight the contribution of *volume* and *sentiment* in $Dataset_1$, and *objectivity* in $Dataset_2$ and $Dataset_3$. This can be explained because $Dataset_2$ (technology) and $Dataset_3$ (automotive) are likely to contain more news and articles with few opinions and sentiments than $Dataset_1$.

5.7 Summary

In this paper, we have described an approach to leveraging community detection for people recommendation. Our work emphasizes the use of implicit sentiment analysis in improving recommendation performance. The experimental results reveal the benefits of our approach compared with some state-of-the-art techniques. Such findings enable us to answer the research questions in Section 5.1 as follows:

1. exploiting sentiments within user recommendation may indeed improve the system performance;

2. the best approach observed is a specific combination of sentiment, volume, and objectivity integrated into Weighed Adamic-Adar and Katz tie strength measures;

3. the aforementioned combination are topic dependent. Particularly, the contributions of sentiments are higher for politic-oriented topics instead of automotive and technology.

As future work, we plan to exploit temporal information for understanding the evolution of relationships between users over time. We also plan to further investigate how parameters α, β, and γ shape the formation of the communities. Although our analysis relies on Twitter, we want to deploy SCORES in a wide domain such as movie or news recommendation. Finally, we plan to integrate big data techniques into our system, with the aim of improving its scalability.

Chapter 6

Matrix Factorization Recommender System

Nowadays, the exponential advancement of social networks is creating new application areas for recommender systems (RSs). People-to-people RSs aim to exploit user's interests for suggesting relevant people to follow. However, traditional recommenders do not consider that people may share similar interests but might have different feelings or opinions about them. In this paper we propose a novel recommendation engine, that relies on the identification of semantic attitudes, that is, sentiment, volume, and objectivity extracted from user-generated content. In order to do this at large-scale on traditional social networks, we devise a three-dimensional matrix factorization, one for each attitude. Potential temporal alteration of users' attitudes are also taken into consideration in the factorization model. Extensive offline experiments on different real world datasets, reveal the benefits of the proposed approach compared with some state-of-the-art techniques.

6.1 Introduction

Microblogging platforms are one of the most versatile and popular technologies on the Internet today. For instance, Twitter sees over 500 million microposts (or *tweets*) published every day on a huge variety of topics, with spikes of more than 100 thousands tweets per second when particular events occur [1]. With the proliferation of user-generated content such as reviews, discussion forums, blogs, and tweets, detecting sentiments and opinions from the Web is becoming an increasingly widespread form of data interpretation. In particular, sentiment analysis aims to understand subjective information, such as opinions, points of view, and feelings expressed by users in the contents they generate.

[1] https://blog.twitter.com/2013/new-tweets-per-second-record-and-how

People-to-people recommendation is an important application in these platforms. Almost all the services are capable of recommending interesting users to follow. However, this recommendation task is not easy due to huge graphs of social ties and fast changing contents that must be analyzed. In this scenario, simple people recommendation algorithms based on content similarity and popularity paradigms are usually considered, at the expense of the recommendation accuracy.

In this paper, we propose a novel people-to-people recommender system that takes into account the users' attitudes towards discussed topic. The proposed recommender enables us to leverage users' attitudes such as sentiment, volume, and objectivity extracted from the semantics of tweets, define a *sentiment-volume-objectivity (SVO)* function, and exploit such knowledge to suggest relevant people to follow. The rationale behind this work is that people might have similar interests but different opinions or feelings about them. Therefore, considering the contribution of users' attitudes may yield benefits to people recommendation. For example, two users interested in the topic "Hillary for President" are likely to be friends. However, the two users may exhibit the same (both support or oppose Hillary Clinton) or contradictory (one supports and the other opposes) sentiments. We are, therefore, supposing that the two users are more likely to become friends in the former case than in the latter.

To handle large-scale social networks, we model this recommendation task using matrix factorization techniques in four steps: (i) build a three-dimensional matrix in which each dimension is represented by a SVO user feature; (ii) learn a latent embedding space from the user-attitudes matrix; (iii) compute the user-user similarity by taking into account the three matrix dimensions; (iv) recommend to a target user a list of relevant people to follow.

In this work, we address two research questions that arise when approaching the people-to-people recommendation problem:

1. Does content published by users and, in particular, the inferred attitudes, allows for a better identification of potential relationships between users?

2. How does temporal analysis of these attitudes impact the accuracy of the recommendation?

The scientific contributions coming from this paper are: (i) an algorithm for people-to-people recommendation on microblogging platforms that takes advantage of features that represent the users' attitudes on specific topics; (ii) a comparative experimental results of a set of different evaluation metrics, including a range of non-accuracy measures, such as diversity and novelty; (iii) a proof of how the recommendation accuracy can be improved

by taking into account the temporal variations of the attitudes expressed by the user; (iv) an evaluation of the proposed algorithm on real world datasets, showing that the considered users' attitudes have unequal correlation with respect to the accuracy of the recommendation, and strongly depend on the topic under consideration.

The rest of the paper is organized as follows: Section 6.2 introduce the problem formulation. Section 6.2.3 describes the recommendation algorithm. Section 6.3 presents the performed experiments to evaluate the proposed strategy and outlines main results. Section 6.4 contains a description of some state-of-the-arts approaches. Section 6.5 contains comment, and conclusions.

In this section, we provide the definition of the people-to-people recommendation problem.

Let $\mathbb{U} = \{u_1, \cdots, u_N\}$ represents the set of users with a valid account on the microblogging platform. In our scenario, an adjacency matrix $A^{N \times N}$ represents the explicit ties, where each element $A_{i,j}$ denotes if the user u_i *follows* (or is *friend of*) the user u_j or not, and therefore is usually expressed by a binary value $\{0, 1\}$. Then, let $\overline{\mathbb{U}} = \{u_1, \cdots, u_M\}$ represent the set of candidate users $u_j \in \mathbb{U}$ without an explicit tie with the target user u_i, that is,

$$\overline{\mathbb{U}} = \{\forall u_j \in \mathbb{U} \mid i \neq j \wedge (A_{i,j} = 0 \wedge A_{j,i} = 0)\}$$

Under this setting, the problem can be formulated as follows: given the matrix $A^{N \times N}$, which represents a known set of social relations between N users, define the following function r:

$$r : \mathbb{U} \times \overline{\mathbb{U}} \rightarrow [0, 1] \tag{6.1}$$

such that, given a *target* user u and an adjacency matrix, returns a value between 0 and 1, which expresses the relevance degree of the candidate user u_j for the target user u_i. Based on such value, the system provides the target user with a recommendation list of the top relevant candidates.

First attempts to people-to-people recommendation take advantage of global models and collective classification for the definition of the r function. In other words, they operate on the whole graph of related nodes rather than deriving individual structural and content-based attributes. The problem is therefore seen as the optimization of one global objective function.

Since *link prediction problem* [115, 173] aims at inferring future interactions and missing links on large graphs, various predictors based on the interpersonal social structure (e.g., common neighbors predictor) are also considered for the ranking task.

Figure 6.1: Principal steps for the people-to-people recommendation task.

Our goal is to define the function r by extending the recommendation analysis to relevant information associated with users that can be retrieved by the micro-blogging platform, namely, the timeline consisting of sequences of microposts. In the rest of the paper, we indicate with $\mathbb{T}$ the set of potential microposts that can be published and with $T_u \subset 2^{\mathbb{T}}$ the most recent microposts published by the user u.

6.2 The proposed people-to-people recommendation

In this section, we introduce our method for recommendation. A strong correlation exists between the presence of a social tie between two users and the topical similarity of explicit activities of these users in the network [11]. So it is logical investigating the chance of predicting the presence of a tie based on user profile features. The idea behind the proposed approach is that, by taking into account the attitudes, in terms of manifested expressions of favor or disfavor on specific matters, the accuracy of the people-to-people recommender is improved. Multiple steps are demanded to implement the recommendation task, as shown in Fig. 6.1.

The timeline of users $u_i \in \mathbb{U}$ are first retrieved. A traditional pre-processing of microposts simplifies the identification of relevant features. All characters are converted to lowercase letters and retweet designations (e.g., "RT"), citations, and URLs are removed. Then, text is tokenized into keywords, from which a list of unigram features is created. Traditional stopwords are excluded from the lists.

Micro-blogging services allow users to include metadata tags in the form of keywords followed by the hash symbol #, which are referred to as *hashtags*. By including them in the posts, the author is suggesting them as good candidates in quality of search keys. Popular hashtags often refer to topics that most people are interested in, including breaking events and persistent discussions [39]. For this reason, they are often considered for clustering posts related to specific topics [151, 36].

Let $\mathbb{C}$ denote the set of all possible concepts. Given a micropost τ, we indicate with $\tau^{(\mathbb{C})}$ the subset of concepts $\mathbb{C}$ that are included in τ, identified by extracting the hashtags in τ. By extension, $T_u^{(\mathbb{C})}$ is the set of concepts that are included in the user u's timeline. The so-obtained representation of microposts is subjected to the SVO analysis (see Sects. 6.2.1

and 6.2.2), which aims at determining the user's attitude on each topic. Since determining similarities among users who have limited activity on specific topics is a challenging task, the SVO-based analysis is not performed on concepts not appearing in a timeline above a given frequency threshold (i.e., 10 tweets). This procedure is commonly followed when attitudes expressed by large audiences are explored [114].

Each user's timeline is subjected to a text categorization process based on a Support Vector Machine (SVM) algorithm [93], so that one or more categories belonging to the set $\mathbb{K}$ of all possible macro-categories are associated to the user according to the published content. These macro-categories (namely, *world*, *elections*, *business*, *technology*, *entertainment*, *sports*, *science*, and *health*) are similar to the ones of a popular online news aggregator [66]. The training set is built-up by retrieving titles and snippets of each macro-category on the aggregator over a period of one month. We denote with $T_u^{(\mathbb{K})} \subseteq \mathbb{K}$ the macro-categories assigned to the user u.

When the system returns a ranked list of people to follow, the target user u's latent factors are compared with the ones obtained from all users that have debated similar topics. This latter *candidate set* $\overline{\mathbb{U}}$ is built-up from $\mathbb{U}$ as follows:

$$\overline{\mathbb{U}} = \{u' \in \mathbb{U} \mid T_u^{(\mathbb{C})} \cap T_{u'}^{(\mathbb{C})} \neq \emptyset \wedge T_u^{(\mathbb{K})} \cap T_{u'}^{(\mathbb{K})} \neq \emptyset\} \tag{6.2}$$

so that, the overlap between u and a candidate u' is extended to the set of macro-categories assigned to each user. Details on the implementation of the r function that assigns a rank to each candidate user can be found in Sects. 6.2.3 and 6.2.4, whereas the following two sections detail the identification of users' attitudes.

6.2.1 Sentiment analysis of microposts

Sentiment analysis or opinion mining is formally defined as the computational study of user's attitudes about an entity expressed in a text [116]. Sentiment analysis is a complex task, hence some assumptions are needed. There are multiple granularity levels of sentiment analysis, as explained in [9]: feature-level, entity-level, sentence-level, document-level. Given the limitations of the micropost length (i.e., 140 characters), we consider sentiment analysis at sentence-level, which corresponds to a whole micropost in our domain. Formally, the goal of our sentiment analysis is to define the following function:

$$sa : \mathbb{T} \rightarrow \{s^{(+)}, s^{(-)}, s^{(\emptyset)}\} \tag{6.3}$$

where the output is composed of three symbols referring to positive, negative and neutral sentiment expressed by the given micropost.

Several approaches have been proposed for the implementation of this function [116] with an average accuracy from 70% to over 82% by means of techniques based on Naïve Bayes (NB) classification [65], a simple model which provides high performance on text categorization. To solve this sentiment analysis task, we devise a multinomial NB model that takes into account multiple features such as (i) unigram features extracted from each post, (ii) negation cues as proposed in [37], (iii) words polarities using the LIWC dictionary [2], and (iv) a part of speech tagger provided by Stanford University [3]. Furthermore, a feature selection based on the salience and entropy measures has also been considered to improve the accuracy of the classifier by filtering less relevant keywords [137]. Maximum likelihood estimate is finally employed for the parameter estimation, with add-1 smoothing utilized for unseen features.

6.2.2 SVO-based analysis

User u's attitudes toward a given topic are evaluated from the observable activity and its aspects. In the micro-blogging scenario, we aim at representing attitudes towards each concept $c \in T_u^{(\mathbb{C})}$ through the following three factors: sentiment, volume, and objectivity. *Sentiment* represents a feeling or opinion about a concept expressed by the user, and is obtained as follows:

$$f_u^{(s,c)} = norm\left(\frac{n_u^{(c,+)} - n_u^{(c,-)}}{n_u^{(c,+)} + n_u^{(c,-)}}\right) \tag{6.4}$$

with

$$n_u^{(c,+)} = \left|\{\forall \tau \in T_u \mid s(\tau) = s^{(+)} \wedge \tau^{(\mathbb{C})} \cap \{c\} \neq \emptyset\}\right| \tag{6.5}$$

$$n_u^{(c,-)} = \left|\{\forall \tau \in T_u \mid s(\tau) = s^{(-)} \wedge \tau^{(\mathbb{C})} \cap \{c\} \neq \emptyset\}\right| \tag{6.6}$$

where $n_u^{(c,+)}$ and $n_u^{(c,-)}$ are the sums of the positive and negative posts, respectively, written by the user u regarding the concept c. Since the range of values can vary widely, the *norm* function scales the values within the [0, 1] and takes on the following expression:

$$norm(x) = \frac{1}{1 + 10^{-x}} \tag{6.7}$$

The second attribute is *volume* and indicates how frequently the user discusses a concept, and is defined as follows:

$$f_u^{(v,c)} = \frac{n_u^{(c)}}{n_u} \tag{6.8}$$

[2] `http://liwc.net` (last visited on 20 December 2016)
[3] `http://nlp.stanford.edu/software/tagger.shtml` (last visited on 20 December 2016)

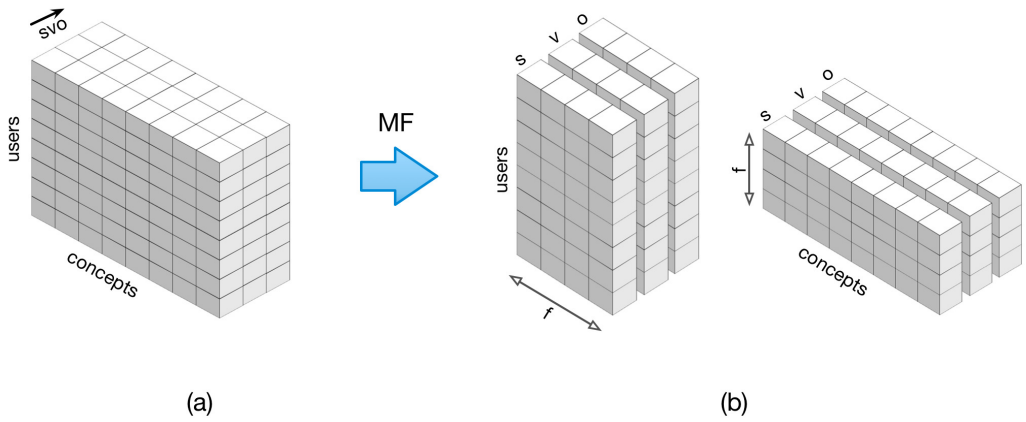


Figure 6.2: The initial user-concept matrix (a), and the matrices representing the correlation between users, concepts and the latent factors (b).

where

$$n_u^{(c)} = \sum_{\tau \in T_u} \left| \tau^{(\mathbb{C})} \cap \{c\} \right|, \quad \text{and} \quad n_u = \sum_{\tau \in T_u} \left| \tau^{(\mathbb{C})} \right| \tag{6.9}$$

The final contribution is *objectivity*, which expresses how many posts about a concept do not contain any positive or negative attitude. It is defined as follows:

$$f_u^{(o,c)} = \frac{n_u^{(c,\emptyset)}}{n_u^{(c,+)} + n_u^{(c,-)} + n_u^{(c,\emptyset)}}$$

where

$$n_u^{(c,\emptyset)} = \left| \{ \forall \tau \in T_u \mid s(\tau) = s^{(\emptyset)} \wedge \tau^{(\mathbb{C})} \cap \{c\} \neq \emptyset \} \right|$$

where $n_u^{(c,\emptyset)}$ is the sum of posts without positive or negative attitudes written by the user u concerning the concept c. We are now able to introduce the SVO vector for the user u and concept c, which takes into account the three factors as follows:

$$\overrightarrow{SVO_u^{(c)}} = [f_u^{(s,c)}, f_u^{(v,c)}, f_u^{(o,c)}] \tag{6.10}$$

6.2.3 Matrix factorization model

Matrix Factorization (MF) techniques [107] are employed for learning the latent characteristics of users and concepts, and defining an approximation of the r function (see Eq. 6.1) by modeling the ranking with inner products in that latent space. The goal is factorizing a 2-dimensional matrix into two matrices $P \in \mathbb{R}^{|\mathbb{U}| \times f}$ and $Q \in \mathbb{R}^{|\mathbb{C}| \times f}$ such that PQ^T approximates the initial matrix, that is, minimizes a loss function between observed and predicted values. Each row q_i represents the association strength between a user and the latent characteristics. Similarly, each row p_j represents the strength between a concept and the latent

dimensions. In the case of micro-blogging platforms, where the number of users and concepts can be very high, this form of decomposition model allows us to keep bounded the storage requirements by tuning the parameter f (i.e., the number of latent factors) accordingly. In our approach, the SVO-based analysis determines a 3-dimensional vector associated to a pair $(user, concept)$, where the concepts are obtained by analyzing the recent activity on the user's timeline. The observed data forms a ternary relation between users, concepts and SVO features, so we obtain a 3-dimensional sparse matrix $M \in \mathbb{R}^{|\mathbb{U}|\times|\mathbb{C}|\times 3}$, as shown in Fig. 6.2(a).

Tensor matrix factorization is a generic model framework for recommendations that is able to handle multiple dimensional data taking advantage of the matrix factorization models [99]. Due to multi-dimensional input data, tensor MF seems to be the perfect choice for the dimension reduction task. In our scenario, as proven in Section 6.3.3, the SVO components representing the user's attitudes have different relevance in the recommendation process according to the category of topics under consideration. For this reason, we decide to perform three MF models, each associated with one of the SVO components, keeping the recommendation process distinct w.r.t each component. For the sake of clarity, we indicate with $P^{(s)}$, $P^{(v)}$ and $P^{(o)}$ the three matrices obtained by the MF model considering the S, V and O component of the SVO vector, respectively, and similarly, we obtain three matrices $Q^{(s)}$, $Q^{(v)}$ and $Q^{(o)}$. Below, we formalize the computation of the only S component, since the other two assume similar formalism. The matrices $P^{(s)}$ and $Q^{(s)}$ are determined by minimizing the regularized squared error:

$$min_{p^{(s)*},q^{(s)*}} \sum_{j=0}^{|\mathbb{U}|} \sum_{i=0}^{|\mathbb{C}|} (M_{i,j}^{(s)} - p_j^{(s)^T} q_i^{(s)})^2 + \lambda(\left\|p_j^{(s)}\right\|^2 \left\|q_i^{(s)}\right\|^2) \tag{6.11}$$

where $M_{i,j}^{(s)}$ is the (i, j) value considering the s attitude, the regularization factor λ is fixed to 0.1, and the summation is extended only to the concepts on which the user u_i has expressed an attitude, that is, $M_{i,j}^{(s)}$ is known. An iterative approach based on the *alternating least squares* technique with regularization [90] is adopted for ensuring the convergence of the Eq. 6.11, that is, when either the matrices P and Q are no longer changing or the change is not significant. One of the strengths of this optimization technique is its ability to handle large sparse datasets built up of implicit interactions between users and items. Moreover, parallel implementations suitable for distributed processing frameworks are also available (see, for instance, [196, 190, 179]).

Now, each user $u_i \in \mathbb{U}$ is associated with a vector $q_i^{(s)} \in \mathbb{R}^f$. The rating of the candidate user u_j to be considered for recommendation to u_i is predicted by the cosine similarity measure

as follows:

$$r_{i,j}^{(s)} = \frac{q_i^{(s)} \cdot q_j^{(s)}}{\left\|q_i^{(s)}\right\| \left\|q_j^{(s)}\right\|} \tag{6.12}$$

The contribution of the three components SVO is linearly combined, as follows:

$$r_{i,j;k} = \alpha_k^{(s)} r_{i,j}^{(s)} + \alpha_k^{(v)} r_{i,j}^{(v)} + \alpha_k^{(o)} r_{i,j}^{(o)} \tag{6.13}$$

where $\alpha_k^{(s)}$, $\alpha_k^{(v)}$, and $\alpha_k^{(o)}$ are three constants in the [0, 1] interval and depend on the macro-category k under examination. Section 6.3.3 describes the procedure to estimate these parameters.

As mentioned in Section 6.2.3, the candidate set of users $\overline{\mathbb{U}}$ consists of the users $u_j \in \mathbb{U}$ that have discussed topics similar to those discussed by the target user u_i. Since the categorization may assign more than one macro-category in K for each pair of users (i, j), multiple $r_{i,j;k}$ values have to be combined. As a result, we select the highest ranking among the considered macro-categories as follows:

$$r_{i,j} = max_{\forall k \in \mathbb{K} | k \in T_{u_i}^{(\mathbb{K})} \wedge k \in T_{u_j}^{(\mathbb{K})}} \; r_{i,j;k} \tag{6.14}$$

The "Who to follow" functionality in microblogging platforms is often implemented with a list of users that does not depend on the current submitted query or context, as in the case of Twitter [169]. So it seems rational to collect the users that show any form of content-based similarity with the target user, with no regard to a specific macro-category.

As with ratings in collaborative filtering approaches, potential *bias* may exist in terms of both attitudes expressed by users and average perception of debated concepts. Two users might be debating on the same concept, but one being a cynic who expresses often negative attitudes, and the other showing a more enthusiastic disposition. In addition, selected topics on micro-blogging platforms might enjoy strong popularity due to several reasons. In this scenario, the popularity bias usually denotes the tendency for some items to be recommended more frequently [31]. Other forms of bias may generate variations in the attitudes expressed by the user on particular concepts. User's bias corresponds to that tendency of the user to give better or worse ratings than the average.

Koren [105] proved that, by considering user and concept biases in the recommendation, improvements can be obtained because it can allow for the intrinsic difference between users and the between concepts to be represented. MF models face these effects by explicitly taking into account the bias parameters as follows:

$$b_{i,j}^{(s)} = \mu^{(s)} + b_{u_i}^{(s)} + b_{c_j}^{(s)} \tag{6.15}$$

where the terms b_{u_i} and b_{c_j} represent the observed deviations of user u_i and concept c_j from the average values, and μ is the overall average value of the s dimension. They describe general properties of users and concepts, without accounting for any involved interactions. These bias parameters are summed up with the predicted ranking $p_j^{(s)^T} q_i^{(s)}$ during the minimization phase obtaining:

$$min_{p^{(s)*},q^{(s)*}} \sum_{j=0}^{|\mathbb{U}|} \sum_{i=0}^{|\mathbb{C}|} (M_{i,j}^{(s)} - \mu^{(s)} - b_{u_i}^{(s)} - b_{c_j}^{(s)} - p_j^{(s)^T} q_i^{(s)})^2 + \lambda (b_{u_i}^{(s)^2} + b_{c_j}^{(s)^2} + \left\| p_j^{(s)} \right\|^2 \left\| q_i^{(s)} \right\|^2) \quad (6.16)$$

6.2.4 Temporal analysis of attitudes

User's attitudes constantly change over time, thus tracking the temporal dynamics of user's interests may help improve personalized systems. The proposed MF-based recommendation includes static representations of interests and concepts. A possible solution is to extend the model by considering potential evolution of these two dimensions over time.

Each timeline T_u is partitioned into $N_{\Delta t}$ intervals of Δt time span. The SVO-based analysis required for the definition of the matrix M is performed on each of these intervals. Therefore, we obtain multiple matrices, one for each time span, on which we perform the MF. The rationale is that, given two users, if they both have discussed the same topic but at different times, they have to be considered less relevant to each other than users that have discussed same topics at similar times. Formally, each rating function $r_{i,j}$ is dependent on the time slot $t_l \rightarrow t_l + \Delta t$ with $l = [1, \cdots, N_{\Delta t} - 1]$, as well. The final ranking is obtained by averaging the time-dependent ranking as follows:

$$r_{i,j} = \frac{1}{N_{\Delta t}} \sum_{l=1}^{N_{\Delta t}-1} r_{i,j}(t_l) \quad (6.17)$$

where $r_{i,j}(t_l)$ is evaluated by considering the partition of the users' timeline in the interval $t_l \rightarrow t_l + \Delta t$.

6.2.5 Computational Complexity

The computation complexity of the approach is driven by the MF process. Indeed, in order to provide up-to-dated recommendations, the MF must be regularly recomputed according to new published content. Instead, the complexity of the SVO-based analysis is determined by the SVM-based categorization of microposts to pre-defined classes (Sect. 6.2.3) and NB classification used for the sentiment analysis (Sect. 6.2.1), which can be trained at once, so we are more interested in the computational requirements after the training step.

Table 6.1: Statistics of datasets.

	Time span	# Tweets	# Users	Lang	Topic
D1:	Jan 2013 → Feb 2013	1,0M	71K	IT	Political Elections
D2:	Sep 2015 → Feb 2015	3,5M	181K	EN	Car Brands
D3:	Dec 2014 → Feb 2015	2,9M	110K	EN	Mobile Phone Brands
D4:	Jan 2015 → Dec 2015	1,2M	99K	IT	Movies
D5:	Jan 2016 → Mar 2016	25,3M	1,1M	IT	Trending Topics

A popular algorithm that implements non-negative MF [112] has computational complexity $O(f|\mathbb{C}||\mathbb{U}|)$ per iteration during the training phase. Of course when new information is added to the user-concept matrix, the factorization can be initiated from the P and Q matrices obtained in the previous cycle, speeding up the time required for the completion of the iterative process to the convergence of the Eq. 6.11. Once the MF is completed, the rating for a candidate user is computed with $O(f|\mathbb{C}||\mathbb{U}|)$ complexity, where f is related to the computation of the cosine similarity (Eq. 6.1), and $|\mathbb{C}||\mathbb{U}|$ is due to the retrieval of the users whose timelines contain hashtags similar to the ones in the target user's timeline. A pre-processing of the set $\mathbb{C}$ removes from the feature space those hashtags whose micropost frequency is less than some predetermined threshold. The assumption is that rare hashtags are non-informative for the recommendation.

Since TDMF takes into consideration a constant number of partitions of the timeline, the above-mentioned big O notation is still valid but a $N_{\Delta t}$-fold increase exists in the processing time.

As for the SVO-based analysis, the categorization based on the SVM technique shows complexity of $O(|\mathbb{V}|)$, where $\mathbb{V}$ corresponds to the vocabulary of terms that compose $\mathbb{T}$. $O(|\overline{\tau}|)$ is the complexity of the NB classification of a timeline's micropost to one of the three classes $\{s^{(+)}, s^{(-)}, s^{(\emptyset)}\}$, where $|\overline{\tau}|$ is the average length of a post. The two computations are performed for each recent post in the user's timeline T_u, so the SVO-based analysis shows an approximate complexity of $O(|\mathbb{U}||\mathbb{V}|)$ by considering the number of posts and the average length of a posts constant.

6.3 Evaluation

Experimental tests of the proposed approach were performed on different real-world datasets, obtained by monitoring the traffic produced by users on Twitter. Such data enabled us to realize a comparative analysis of our system with similar approaches proposed in the research literature.

To guarantee a correct statistical significance of the results, the experimental evaluation were carried out taking into account different datasets as shown in Table 6.1. The considered datasets were gathered as follows:

D1: We filtered from the Twitter stream the hashtags related to politician leaders and Italian parties during the 2013 Italian general election.

D2: A dataset on majors technology brands, among others Samsung, Apple, Nokia, Huawei, LG, Motorola, and Blackberry.

D3: Tweets matching terms related to the automotive landscape, such as Audi, BMW, Ferrari, Jaguar, Mercedes, Toyota, and Porsche.

D4: Corpus of tweets that counts more than 200 movies released in Italy during 2015.

D5: This dataset includes tweets of trending topics automatically suggested by the microblogging platform over a period of 3 months, such as #bruxellesattacks, #oscars, #syriaconflict.

The time period in which each dataset has been collected is splitted in two parts. The initial 70% corresponds to the data for the training set, the subsequent remaining data are used for testing the proposed recommendation system against other benchmarks. A crawler periodically updated the profiles of each user during the whole time period, considering also new followers/following relationships. Each time a social tie is discovered, its timestamp is being associated with the time the crawler found it.

6.3.1 Benchmark: metrics and comparative algorithms

A wide spectrum of evaluation metrics for RSs exist, most of them focused on their accuracy. While the assessment of such aspect is foundamental, there are limits that emerge due to the discrepancy between the users' perception and the outcome of the metrics [123]. An accurate recommendation, however, is not necessarily perceived as a useful one. If the users in the recommendation list are very similar to the target users, the benefits of the system are limited because good chances are that the users discover them by querying the microblogging service or exploring the neighbors of their personal social network by themselves. A more useful recommender provides accurate and personalized recommendations guaranteeing, at the same time, high levels of novelty and diversity. For this reason, multiple metrics have been considered for better evaluating the optimal trade-off between accuracy, novelty and diversity of the considered recommendation approaches.

Accuracy

The goal of the people-to-people recommendation is to provide the target user with a set of relevant people to follow. In our approach, the output is a set $\mathbb{L}_u$ of potentially relevant users, where the timeline of each u in $\mathbb{L}_u$ is considered for the user profiling.

A straightforward methodology to measure the accuracy of a RS is to assess how many suggestions are relevant to the user. We suppose that user u_i is relevant to u_j if a real *following relationship* exists between them.

Precision is the most used accuracy measure and gives a general idea of the overall performance of the recommendation. Since it is known that users focus their attention on the top ranked items of a list [95], we employed the *Success at Rank K (S@K)* measure that is commonly used for evaluating ranked lists of recommendations. It expresses the mean probability that a relevant user is located in the first K positions of the suggested users set, and is obtained as follows:

$$Acc(u, \mathbb{L}_u) = \frac{1}{|\mathbb{L}_u|} \sum_{u_i \in \mathbb{L}_u} S@K(u_i) \tag{6.18}$$

where $S@K(u_i)$ is one whether u_i is a relevant user for the target user u, zero otherwise.

Diversity

Diversity generally applies to a set of recommended items, and is related to how different the items are with respect to each other. Diversity is then determined by evaluating the dissimilarity of textual features extracted from users' timeline of the recommended set $\mathbb{L}_u$.

The diversity measure we devised is based on the Latent Dirichlet Allocation (LDA) [25], a generative probabilistic model for collections of discrete data such as text corpora. LDA shapes latent topics as a distribution over the words of the vocabulary, and every given document as a distribution over these topics, which is sampled from Dirichlet distributions. LDA model is often used for dimensionality reduction, where any input document d is assigned to a fixed set of real-valued features, that is, the posterior Dirichlet parameters $\gamma^*(d)$. If we assume that γ^* is represented by means of a vector, we define the diversity based on LDA as follows:

$$Div(\mathbb{L}_u) = 1 - \left\| \gamma^*(d(T_{\mathbb{L}_u})) \right\| \tag{6.19}$$

where $d(T_{\mathbb{L}_u})$ represents a text document consisting of the concatenated posts from the users' timeline in $\mathbb{L}_u$. The LDA diversity reaches high values if the combination of users' timeline represent several different latent topic.

Novelty

There have been several attempts to capture the degree of novelty in single measures [130,

41]. Novel recommendations consist in suggesting items that the user did not know about, and whose description is semantically far from users' interests. Therefore, the measure takes into consideration both the recommended content and the target users' interests. Hijikata et al. [87] use collaborative filtering to derive novel recommendations by explicitly asking users which items they already know. The scale of the domain we are dealing with and the number of users involved do not allow us to follow a similar methodology. The novelty measure assumes high values if the recommended users' timeline include several topics that are not discussed yet by the target user. Therefore, we can define novelty in terms of overlap among topics discussed by the target user u and the suggested users $\mathbb{L}_u$. More formally we define:

$$Nov(u, \mathbb{L}_u) = \frac{1}{|\mathbb{L}_u|} \sum_{u_i \in \mathbb{L}_u} \frac{1}{|T_u^{(\mathbb{C})}|} \sum_{c \in T_u^{(\mathbb{C})}} (-\frac{n_{u_i}^{(c)}}{n_{u_i}}) \tag{6.20}$$

6.3.2 Algorithms for comparative evaluation

In order to outline comparative conclusions from the experimental evaluations on the considered datasets, the following people-to-people recommendation approaches have been devised and included in the experimental tests:

R: A baseline recommender that randomly suggests users from the considered dataset.

NP: A non-personalized recommender that always suggests the most popular users in the dataset, that is, the users with the highest number of followers.

CB: The content-based approach proposed in [81] (with the name of `S1`), which represents each user u through the function $d(T_u)$, that is, the text document consisting of the concatenated posts from the users' timeline. A traditional search engine based on the vector space model with a TF-IDF scoring function and cosine similarity measure [19] returns the users that are more similar to the target one by considering their timeline's content.

CF: It represents each user u_i by the following set:

$$\{\forall u_j \in \mathbb{U} \mid i \neq j \wedge (A_{ij} = 1 \vee A_{ji} = 1)\} \tag{6.21}$$

that includes any user with an explicit tie with u_i (i.e., followers and followees). The IDs of these users are converted to unique keywords and, similarly to the CB approach, a IR-based search engine returns a ranked link of recommendations. It corresponds to the `S7` approach in [81].

CBH: Similarly to CB, each user is represented by the posts included in the timeline, but instead of every keyword, the content is limited to the set of concepts in $T_u^{(\mathbb{C})}$. The frequency of the concept in the user's timeline corresponds to the term frequency.

FoF The Friend-of-Friend recommender is available in popular social network services, such as Facebook and LinkedIn [33, 69]. It relies on the following hypothesis: if many users followed by u subsequently follow a particular person, this latter person is more likely to be suggested to u. The greater the number of u's friend that follow the candidate, the higher is the relative rank in the suggested list. It follows the common neighbor paradigm that makes use of explicit social ties often considered in the link prediction task [115].

MFE: The straightforward recommender based on MF [107] where the items to suggest are the users themselves. Therefore, the training set is composed of ratings $r_{u_i,u_j} \in 0, 1$, which represent the existence of an explicit social tie (i.e., following relationship) that bind the pair of users. The estimated rating between the target user u_i and the generic candidate u_j is obtained by the inner product in the latent factor space, that is:

$$q_i^T p_j$$

The top-ranked candidates of the target user are the ones assessed in the evaluation.

MF: The recommendation approach based on the SVO-based analysis and the MF models introduced in Section 6.2.

TDMF: The previous recommendation approach enhanced with temporal dynamic features, as explained in Section 6.2.4.

The explicit social ties used by CF, FoF and MFTB approaches are extracted from the training set, whereas the test set is used to assess the performances. Similarly, the timelines considered for the learning process in the CB, CBH, MF and TDMF approaches consist of microposts published in the first split (i.e., training set) of each dataset.

6.3.3 Experimental results

The evaluation of the accuracy is achieved by comparing our system with some state-of-the-art people-to-people RSs. To perform an offline comparison analysis, an evaluation set has been built. We selected 1,000 random users from each of the dataset already introduced in Section 6.3, that match the following criteria:

- users that posted at least five tweets
- users with at least ten friends and followers into the dataset (that can be selected for the evaluation test)

This kind of offline analysis suffers from an important weakness: the natural sparsity of datasets derived from social network limits the amount of relevant content that can be evaluated. In this way, selecting exclusively random users without matching the above-mentioned criteria may lead to have no real friends or followers to compare with into the test dataset, and therefore resulting in a zero accuracy for every recommender. On the other hand, offline evaluations are often considered in RS studies because they allow researchers to perform large scale evaluations on thousands of users, different datasets and algorithms at once [86].

In Table 6.2, we report the results of the comparative analysis in terms of accuracy. All the experimental results were tested for statistical significance through a two-tailed *t-test* with a significance level set to $p < 0.05$.

In terms of accuracy, the outcomes show the substantial benefits obtained with the proposed approaches and confirm our initial hypothesis about the potential combination of sentiment, volume and objectivity to better identify real relationships between users. A traditional MF-based approach that limits its analysis to the explicit ties between users, i.e., **MFE**, does not reach similar accuracies. The results highlight also how the **TDMF** approach obtains the best values among all datasets. This is a relevant achievement that endorses how important is to consider temporal features for the people-to-people recommendation.

Subsequently, we evaluated the performance of the RSs in terms of diversity and novelty. Table 6.3 summarizes the diversity and novelty obtained on average among all datasets. Approaches that leverage social network information such as **NP**, **CF**, and **FoF** reach high values of novelty, that is, they are able to suggest people that are more likely to discuss topics unknown to the target user. On the contrary, **MF** and **TDMF** techniques, thanks to matrix decomposition and temporal analysis, supply the RS with the ability of suggesting diverse users to follow, that is, a list of recommended users that are different, one from each other.

As for the temporal factor, we analyzed the variation of the accuracy as a function of the extent of the Δt time span. Table 6.4 shows that datasets D1, D2, D3, and D4 achieve the best accuracy with Δt intervals of 14 days and 21 days, while D5 with Δt of 7 days. Since the latter dataset consists of several fragmented and temporary trending topics, by considering a time span of 7 days, the most relevant topics are better represented.

Table 6.2: A comparison of accuracy outcomes among some state-of-the-arts recommender approaches.

RS	D1	D2	D3	D4	D5
R	0.049	0.057	0.024	0.045	0.038
NP	0.146	0.114	0.122	0.111	0.065
CB	0.130	0.118	0.115	0.126	0.111
CF	0.172	0.163	0.161	0.167	0.151
CBH	0.127	0.099	0.105	0.055	0.078
FoF	0.165	0.155	0.159	0.140	0.132
MFE	0.121	0.105	0.111	0.119	0.107
MF*	0.187	0.181	0.178	0.201	0.182
TDMF**	**0.212**	**0.233**	**0.235**	**0.241**	**0.255**

(*) With the best SVO values for each dataset and $f = 5$

(**) Best Δt for each dataset showed in Table 6.4 and $f = 5$

One popular example in the dataset is the news about the 2016 Brussels bombings. By considering a shorter time span, the recommendation is more tailored to users that are interested in the terrorism attack instead of considering people fascinated by the capital of Belgium, its history or cultural events.

In order to understand the behaviour of the users' attitudes, we performed a sensitivity evaluation of SVO parameters through a large-scale *gradient descent algorithm* [192] with learning rate $\zeta = 0.1$. This evaluation enabled us to observe how the performance could be improved by tuning the weights that define the different contributions of sentiment, volume, and objectivity, based on the nature of topics (on which the users' similarity is computed). In particular, the results in Table 6.5 highlight how the contribution of sentiment is higher for topics about politics and movies, while the contribution of volume is on average significant for all of the considered topics.

Finally, Figure 6.3 reports the RS accuracy for the **MF** approach as a function of the latent factor's number f. As can be noted, there are no relevant accuracy improvements by increasing the number of latent factors. This finding motivated us to select a fixed f=5 for all of the aforementioned experimental evaluations. A lower number of latent features decreases a lot the computational resources.

The obtained outcomes pave the way to the hypothesis that a hybrid approach that accurately selects the recommendations from multiple approaches, such as **FoF**, **CF** and **TDMF**, may show benefits to the user. For instance, the approach based on explicit social ties (FoF) outperforms attitudes when the goal is to have high novelty, whereas **MF** and **TDMF** obtain in general better accuracy and diversity on the considered datasets. But a simple linear combination of the outputs would not be optimal. Future work is required to

Table 6.3: Results for diversity and novelty metrics

RS	Novelty	Diversity
R	0.21	0.14
NP	**0.29**	0.34
CB	0.14	0.23
CF	**0.39**	**0.55**
CBH	0.11	0.15
FoF	**0.35**	0.29
MFE	0.25	0.28
MF	0.19	**0.45**
TDMF	0.25	**0.47**

Table 6.4: Results of S@10 for TDMF recommender system while varying the length of Δt time span

Dataset	7gg	14gg	21gg	30gg
D1	0.191	0.202	**0.212**	0.187
D2	0.210	**0.233**	0.221	0.200
D3	0.201	**0.235**	0.18	0.199
D4	0.192	0.205	**0.241**	0.225
D5	**0.255**	0.189	0.188	0.173

Table 6.5: Sensitivity analysis of sentiment-volume-objectivity parameters for the best obtained values of MF recommender system

Dataset	S@10	S	V	O
D1	0.187	**0.45**	0.45	0.10
D2	0.181	0.20	**0.60**	0.20
D3	0.178	0.30	**0.65**	0.05
D4	0.201	**0.45**	0.45	0.10
D5	0.182	0.20	**0.70**	0.10

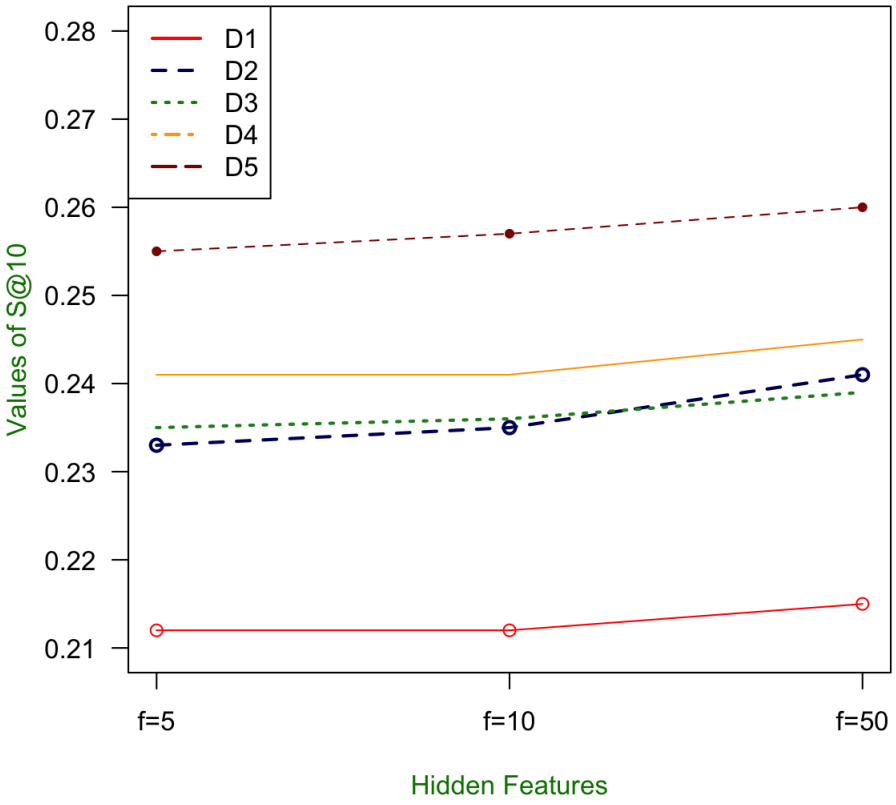


Figure 6.3: S@10 while varying the number of latent factors f, with the best values of Δt

understand what the user is currently expecting from the recommender, promoting items that are not similar to what they have previously liked (i.e., maximizing the diversity), or pursuing higher accuracy, that is, items similar to what users have previously liked.

6.4 Related work

In this section we describe several works somehow related to the proposed system, especially focusing on people-to-people recommendation.

From the seminal works on link prediction [115, 85], many relevant contributions have been proposed. Freyne *et al.* [56] provide the active user with suggestions about key people to connect to, based on social relationship information coming from different external sources and gathered through the social aggregator Sonar [75]. In [34] techniques that exploit both the user-generated content and the social network structure are proposed for recommending people of potential interest to the target user. Such techniques rely on the

Friend-of-Friend (FoF) hypothesis that if many of the target user's friends have a friend in common, this latter could be friend of the target user as well. This system is one of the baseline approaches that appear in the comparative analysis reported in Section 6.3. The authors of [77] address the same problem in an enterprise scenario. They aggregate information from different sources in order to profile users, thus being able to identify those who have provided a similar contribution (e.g., co-author papers, patent authorship, etc.). This work is based on the assumption that if two users have generated content on similar topics, they are more likely to appreciate getting in touch with each other than other users. Quercia and Capra propose a mobile application that relies on the users' physical proximity for generating people-to-people recommandations [146]. In [191], a supervised machine-learning approach is proposed to address the link recommendation problem on an enterprise social network. To this end, the authors mine the user-generated content, the social graph, and the company's organizational chart to profile enterprise users. Some work has been focused on the user recommendation problem in social micro-blogging services like Twitter. In particular, the authors of [82] make a comparison between content-based and collaborative filtering approaches for user profiling. To this end, they resort to a classic search engine to index and classify such profiles via the traditional TF-IDF approach of Information Retrieval. Then, the top-k users are suggested to the target user. Their experimental results show the better performance of collaborative filtering approaches compared to those of content-based.

Such findings suggest that the relations between users are more structured, and therefore more relevant for user recommendation task than the noisy microposts. Given the relevance of these approaches, both of them have been implemented and compared with the proposed system (see Sect. 6.3). In [14] the authors address the same problem through an algorithm which explores the topology of the social graph in Twitter to locate users to recommend to the active user. This approach extends the well-known item-based recommendation nearest neighbor technique [152] to the user recommendation task. However, the works presented in [107] and [194] show that approaches based on matrix factorization provide better performance than those based on neighborhood techiniques. Such consideration, along with the need to operate on large-scale social networks, inspired our approach.

Matrix factorization techniques have been previously considered in the link prediction problem. In [125] the authors combine explicit and latent features and prove the effectiveness on various datasets. Kutty *et al.* [109] propose tensor space models as a potential framework able to include also additional attributes associated with each user. Other works extend the analysis by considering dynamic interactions, that is, the time in which a tie is created, e.g., [6, 70, 60, 109]. The above-cited works have not been explicitly evaluated on

popular social network services, such as Twitter or Facebook, and do not take into account user attitudes.

Yang *et al.* [186] extends the check-ins left by the users on location-based services with additional features, such as fine-grained user preferences extracted from opinions expressed in user comments. MF techniques are considered for capturing both social and inter-venue influence based on similarity measures between user comments, geo-distance, categories, reviews, etc. Similarly, in [187] the authors use a three-way tensor model User×Keyword×Venue for personalized location ranking.

Although a large number of contributions have been devoted to the people-to-people recommendation issue - to the best of our knowledge - exploiting sentiment analysis of user-generated contents for purposes of community detection and/or user recommendation has not been deeply investigated. Xu *et al.* [183] transform the sentiment-based community discovery into a correlation clustering problem and propose a random rounding algorithm based on semidefinite programming for its solution. In [132] the authors describe an unsupervised approach based a non-parametric clustering algorithm for detecting hyper-groups of communities, called *hyper-communities*, where users share the same sentiments. In [188], the authors extract users' interests from their microposts and identify some sentiment-based features that express the likelihood of two users establishing a relationship (i.e., following each other or mutually mentioning) between them. They also advance a factor graph model including a sentiment-based version of the cognitive balance theory for predicting potential relationships.

As far as we aware, this is the first work combining sentiment analysis and matrix factorization techniques to assist users in locating interesting people.

6.5 Summary

In this paper, we have described a people-to-people recommendation approach for large-scale social networks. Our work emphasizes the use of user's attitudes such as implicit sentiment, volume and objectivity to improve recommendation performance and matrix factorization models to maximize efficiency and scalability. The experimental results showed the advantage of our approach compared with the state-of-the-art techniques. Taking advantage of implicit sentiment related to the users' timeline, enables us to better identify the relationship of interest between users. The experimental evaluation on different datasets has also proved that the SVO factors are influenced by the topics under discussion. When multiple factors obtained from the user-generated content are taken into consideration, an adequate analysis of their relevance in the recommendation process is required. The same

conclusion holds for the time unit considered for the temporal analysis of the expressed users' attitudes.

Chapter 7

A Sentiment-based Youtube Video Recommender

Everyday video-sharing websites such as YouTube collect large amounts of new multimedia resources. Comments left by the viewers often provide valuable information to describe sentiments, opinions and tastes of the users. For this reason, we propose a novel re-ranking approach that considers that information in order to provide better recommendations of related videos. Preliminary experiments indicate an improvement in the rank of the proposed resources.

7.1 Introduction

YouTube is the world's most popular web video community used by 1 billions unique users world wide each month[1]. 4 billions of videos are viewed per day, with 100 hours of new ones uploaded every minute. Sifting through this large repository of multimedia resources poses unique challenges for the user.

The YouTube user interface provides, given the current video *id*, a list of recommendations as shown in Fig. 7.1. YouTube selects those recommendations based on an algorithm that considers signals from a variety of sources including the user's favorite, watched and liked videos [40]. These signals are combined for ranking the list of *related* videos compiled by monitoring what other people usually watch next. By exploring this related-video graph, a candidate list is built. Characteristics about the videos (e.g., views and ratings) and the similarities of the videos with the history of videos watched by the user are combined to rank the candidate resources. A balance between relevancy and diversity across categories guarantees some sort of diversity in the related video list $L_{id}^{(y)} = (l_1, l_2, \ldots, l_n)$. As a result of that approach, the user-generated comments that are shown below the video are not taken

[1] http://www.youtube.com/yt/press/statistics.html (Accessed: 23 January 2015)

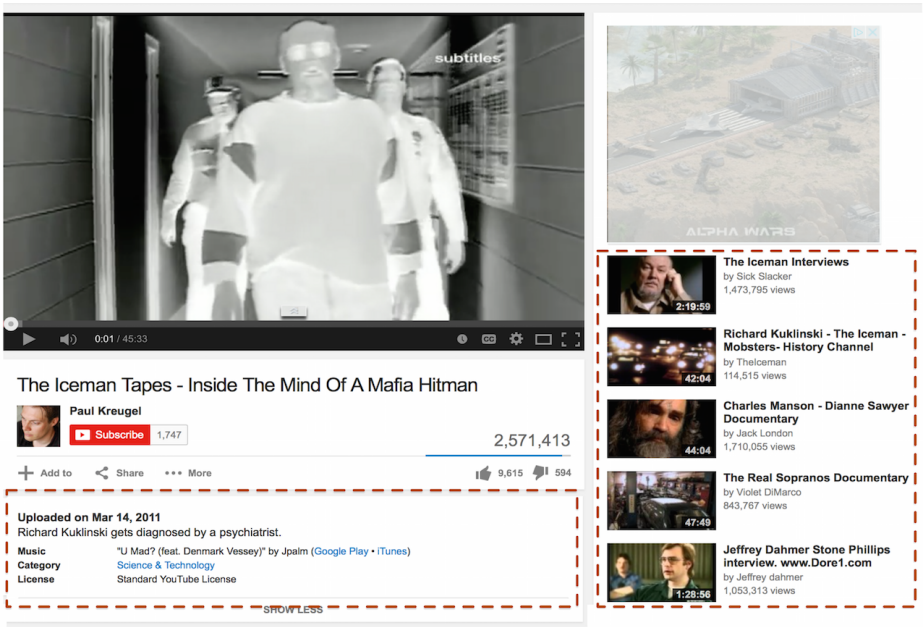


Figure 7.1: The YouTube website with metadata and recommended videos highlighted.

into consideration. While these user interactions are often short and noisy, they have the chance to represent valuable information about user interests, tastes and, more in general, debate topics about the video content.

Consistent with other hypothesis in this thesis, the rationale in this work is that two videos have the chance to be related if they rouse similar reactions and sentiments on the users that watched them. This sort of implicit relationship between multimedia resources might improve the original YouTube ranking in a way that better match the user expectations. While related video lists can host a large number of suggestions (i.e., up to 40), on the flip side videos that are not promoted high on the list are less likely to be visited by the user. For this reason, ranking of related video acquires a paramount relevance.

In this paper we propose a re-ranking method that, for each video, generate an improved order of the resources proposed to the user by the YouTube traditional recommender. Empirical results prove the benefits of including user-generated comments in the recommendation process for real-user scenarios. The rest of the paper is organized as follows: Section 1.2 presents the proposed recommendation model. Section 1.3 highlights the evaluation results, and in Section 1.4 are briefly introduced related work and in Section 1.5 are summarized the results obtained and the research questions answered.

7.2 The Proposed Video Recommendation

Given a video *id*, the YouTube Data API[2] allows us to retrieve up to 1000 comments C_{id} by submitting a sequence of 20 requests, each with a 50-comment limit. The API provides us also the top 25 related videos. We filter too short comments and the ones with swear words. A Bayesian classifier trained on a subset of spam comments help us to filter out the less relevant content.

A straight keyword-based approach [18] identifies the words that express a sentiment, assigning a score to them for each of the following dimensions: *positivity*, *negativity*, and *objectivity*. In particular, given a comment $c_{i,id} \in C_{id}$ we sum up all the positivity scores and then subtract the negativity ones. The obtained real value is compared with 5 intervals so that each comment is assigned to one of the following classes: very positive, positive, neutral, negative, very negative. Those classes are also the five dimensions of a vector space model, where the sentiment vector:

$$v_{id}^{(ss)} = (v_{1,id}, v_{2,id}, v_{3,id}, v_{4,id}, v_{5,id}) \tag{7.1}$$

is calculated summing up the occurrences of the very positive classes for the dimension $v_{1,id}$, positive occurrences for $v_{2,id}$, neutral occurrences for $v_{3,id}$ and so forth.

The same procedure is followed for each video l_j in the related video list $L_{id}^{(y)}$ by analyzing the set of comments associated with j. We obtain n vectors $v_{lj}^{(ss)}$ that can be compared by means of a cosine similarity measure with $v_{id}^{(ss)}$. Therefore, the related video l_j will have a sentiment-based similarity $r_j^{(ss)} \in [0, 1]$.

A second step extracts named entities (e.g., persons, locations) and nouns from each comment by means of a Named-entity recognizer and Part-of-Speech tagger, respectively. As with the previous procedure, two vectors, $v_{lj}^{(ne)}$ and $v_{lj}^{(pos)}$, are obtained for each video j in $L_{id}^{(y)}$ by summing up the contribution of the different comments. The two vectors $v_{id}^{(ne)}$ and $v_{id}^{(pos)}$ are also computed for the *id*-video. The dimensions of the vectors are distinct named entities and nouns that appear in the analyzed user-generated data. A cosine similarity measure assigns the scores $r_j^{(ne)}$ and $r_j^{(pos)}$ between the current video *id* and the j-video, respectively, for the named entity and noun comparisons.

The last step calculates the final rank for the video j by linearly combining the three measures:

$$r_j = \alpha_1 r_j^{(ss)} + \alpha_2 r_j^{(ne)} + \alpha_3 r_j^{(pos)} \tag{7.2}$$

where the three α values are constants that we set to $1/3$ in our experiments.

[2]https://developers.google.com/youtube/ (Accessed: 23 January 2015)

7.3 Evaluation

A total of 8 people were interested in the evaluation, mostly students of computer science courses, all usual users of the YouTube service. A Java application has been developed to assist them during the evaluation. We asked them to select 10 videos $V = (v_1, \ldots, v_{10})$ from their watched history, the recommendation on the YouTube homepage or from the their subscribed channels. For each video v_i, the application obtains its related videos $L_{vi}^{(y)}$ suggested by YouTube. A new rank list L'_{vi} is built by downloading the comments and running the proposed approach on them. At the end, a randomized list is proposed to each user that was asked to evaluate her interests in watching the single videos in it with a five-level Likert scale. The Normalized discounted cumulative gain (nDCG) is evaluated both for the YouTube list $L_{vi}^{(y)}$ and the new ranked one L'_{vi}. After computing the measure for each video we averaged them to obtain an overall performance evaluation. The YouTube recommender obtains a nDCG of 0.829 while the proposed approach reaches 0.858 with an improvement of 3.51% (p-value<0.05).

7.4 Related Works

To the best of our knowledge, our work makes the first attempt to analyze user comments in the video recommendation domain. Shmueli *et al.* [157] analyze users' co-commenting patterns for predicting, for a given user, suitable news stories that she likely comment on. A similar approach is focused on the news recommendation by Messenger and Whittle [126]. Sergiu *et al.* [32] explore the effectiveness of comments and other social signals for the video retrieval task, that is, when a user query must be elaborated.

7.5 Summary

In this work we proposed a Youtube recommender that takes into account of user sentiment within the comments they provide. In a preliminary evaluation, the proposed approach shows that sentiment and semantics in general may help a video recommender. In order to improve the proposed approach, we are currently planning to do an extended evaluation to better understand the relationship between the kinds of opinions and sentiments expressed by the users and the categories of the videos. There are many videos for which YouTube is not able to compute a reliable set of related videos due to the scarcity of user activities. It is interesting to understand if the proposed approach can be successfully implemented even for new videos that have collected a right number of comments, partially addressing

the data-sparsity issue due to the scarcity of user activity records. In addiction, will be evaluated whether the techniques and weighting measure proposed in previous Chapter 4, and Chapter 3, are valuable also in a different social network such as Youtube and in a different recommendation task: from user recommendation to video recommendation.

Chapter 8

Conclusions

With the advent of social media, the continuously growing amount of accessible data and the resulting information overload problem pose new challenges for Recommender Systems.

In this thesis, we described our research work in leveraging Sentiment Analysis methods to build richer user profiles and improve recommendation engine. The main goal was to understand if Sentiment Analysis in social networks can improve the performances of the recommendation process.

8.1 Summary of Contributions

Regarding the research questions that were identified in Section 1, the main findings and contributions of this thesis are summarized below.

Sentiment-based User Recommender on Twitter. As people discuss various topics on microblogging platforms, inferring the sentiment relates to such topics is important to improve user recommendation. The following research question was therefore investigated in this thesis.

- *Can the consideration of a sentiment-based model yield benefits to social RSs?*

To answer this question we conceived a novel sentiment-based function called *Sentiment-Volume-Objectivity* (SVO), related to the user interests. We built a user recommendation engine that uses this weighting function, and collected a dataset for training and testing proposed approach. The devised technique allowed us to build more complete user profiles, and a comparative analysis between the proposed model and some state-of-the-art approaches showed the improvement in terms of recommendation accuracy.

Exploiting Signals and Temporal Dynamics for a People-to-People RS. As users' interests evolve over time, a user recommender can exploit such evolution to perform better suggestions. In this section we pose two research questions as follows:

- Can the consideration of temporal patterns of users' interest really impact the characteristics and quality of user recommender?
- Can Sentiment Analysis yield some benefits to the proposed temporal-based RS?

To answer these research questions we devised a signal-based recommender system, that takes into account of temporal variation of the user interests. In order to analyze such signals we made use of the wavelet transform, a signal processing technique that captures the frequency content of any signal. To evaluate the proposed model we followed over 20,000 users on Twitter and selected a sub-set of 1,500 people for the test process. An extensive comparison with the approach proposed by [81], revealed a positive answer to the first research question. Furthermore, we considered the *SVO* weighting schema presented in Chapter 3 into the signal-based approach, and the evaluation results showed that Sentiment Analysis could improve the recommender precision, also using this kind of method.

Leveraging Community Detection Techniques for User RS. Considering the results of previous chapters that highlight how Sentiment Analysis can improve the recommendation performance, in this section we devised a user recommender that improves recommendation precision also while varying the topic considered in the user profiles. Therefore the research questions we pose are:

- Which is the best graph technique for enhancing the contribution of the Sentiment Analysis in people RSs?
- Can sentiment improves the final recommendation precision?
- Are there differences depending on the category of topics dealt with by the user?

To answer the first question we built a graph-based recommender where nodes are users and weighted ties represent the similarity values between users. Our RS approach considered several tie strength measures to suggest relevant user to follow. The best approach that maximize the RS was a specific combination of sentiment, volume, and objectivity integrated into Weighed Adamic-Adar and Katz tie strength measures. The findings behind the evaluation of those several tie strength measures let us to answer the other two

research questions we posed. Sentiment Analysis also in this case was able to increase the recommender precision, but the aforementioned combination of SVO was topic dependent. Particularly, was discovered that the contributions of sentiments are higher for politic-oriented topics instead of automotive and technology topics.

Matrix Factorization Recommender System. As a matter of fact, the amount and variety of social data is continuously growing in latest years, it is therefore important that current RSs try to handle this huge amount of data. In this section we thus want to answer the following research questions:

- In a scalable RS, does content published by users and, in particular, the inferred attitudes and sentiments, allow for a better identification of potential relationships between users?
- How does temporal analysis of these attitudes impact the recommendation?
- Can sentiment analysis improve also non-accuracy metrics, such as diversity and novelty?

To answer these questions we built a large-scale RS based on a three-dimensional matrix factorization approach, one for each SVO parameter. To the best of our knowledge this was one of the first attempt that combined sentiment analysis, recommender systems and matrix factorization technique. The proposed model enabled us to answer the research questions, that is, (i) taking advantage of implicit sentiment related to the users' timeline, definitively enabled to better identify the relationship of interest between users (ii) considering temporal factors improved the recommender precision, and (iii) the proposed approach increased the recommender diversity while slightly decreased the novelty of the system.

A Sentiment-based Youtube Video Recommender. In order to evaluate a sentiment-based recommender in a different domain such as in a video recommender, we selected Youtube as a data source for our RS. The research question we pose is the following:

- How much Sentiment Analysis can enrich the recommendation process on Youtube?

To answer this research question, we devised a re-ranking system for video recommender that takes into account user's comments and analyses the sentiment within it. The evaluation test obtained with real users, achieved an increment of 3,51% in terms of recommender precision. This preliminary study let us to answer the research question positively, and open a list of future works and research questions also for Youtube social networks.

In summary, this thesis contributes to research regarding sentiment-based RSs in social networks. Each contribution in this thesis confirmed the hypothesis that sentiment is a valuable feature to improve the recommendation process. During the research activities, we devised new recommendation approaches that better integrate Sentiment Analysis algorithms. We built different kind of RS approach, leveraging large-scale techniques such as matrix factorization as well as graph-based techniques. We also exploited signal processing to enhance the contribution of temporal dynamics into the proposed recommender. Finally, we evaluated proposed systems using diverse datasets and social networks sources such as Twitter and Youtube, and considering other state-of-the-arts techniques to have a real benchmark to compare with. Furthermore, many scientific research paper [189, 184, 121, 100, 141, 97] drew inspiration from our work and further confirm that Sentiment Analysis is an effective method to adopt in a Recommender System.

8.2 Future Work

Based on the methods and findings presented in this thesis, we suggest the following recommendations for future work.

The recommender system framework and sentiment techniques presented in this thesis are mainly based on data collected from microblogging systems such as Twitter. User or usage data from other Social Web systems can be further exploited to cover different topics that a user discusses, or different social networks such as Facebook, Instagram, and LinkedIn can be further adopted as a data source for the recommender system. In particular, a user recommender based on Instagram could be proposed, while for Facebook, LinkedIn, and also for Twitter could be possible to realize different kind of RS such as News, Products or Movie recommender. We already did a preliminary attempt to a news recommender using signal-based model [29]. In addiction, other future steps will be the integration of sentiment in this diverse kind of task for RS.

A further possible future development could involve the extraction and classification of other types of concepts in the semantic of text. In addiction to the simple hashtags, it is possible to take into consideration topics, entities, companies, products, and much more types that can become a valuable feature to improve the recommendation process.

Each model proposed in this thesis takes advantage of similar Sentiment Analysis methodology and technique, because we wanted to increase the reproducibility of tests through different works, and have more reliable evaluations as possible. A possible future work will be therefore to analyse different Sentiment Analysis methodologies, such as other

Machine Learning algorithms or a specific technique, and evaluate their effectiveness into the proposed sentiment-based Recommender Systems.

Bibliography

[1] *21st IEEE International Conference on Parallel and Distributed Systems, ICPADS 2015, Melbourne, Australia, December 14-17, 2015*. IEEE, 2015.

[2] Fabian Abel, Qi Gao, Geert-Jan Houben, and Ke Tao. Analyzing temporal dynamics in twitter profiles for personalized recommendations in the social web. In *Proceedings of ACM WebSci '11, 3rd International Conference on Web Science, Koblenz, Germany*. ACM, June 2011.

[3] Fabian Abel, Qi Gao, Geert-Jan Houben, and Ke Tao. Analyzing User Modeling on Twitter for Personalized News Recommendations. In *International Conference on User Modeling, Adaptation and Personalization (UMAP), Girona, Spain*. Springer, July 2011.

[4] Fabian Abel, Qi Gao, Geert-Jan Houben, and Ke Tao. Analyzing user modeling on twitter for personalized news recommendations. In *Proc. of the 19th International Conference on User Modeling, Adaption, and Personalization*, pages 1–12. Springer-Verlag, 2011.

[5] Fabian Abel, Qi Gao, Geert-Jan Houben, and Ke Tao. Semantic enrichment of twitter posts for user profile construction on the social web. In *Proceedings of the 8th extended semantic web conference on The semanic web: research and applications - Volume Part II*, ESWC'11, pages 375–389, Berlin, Heidelberg, 2011. Springer-Verlag.

[6] Evrim Acar, Daniel M. Dunlavy, and Tamara G. Kolda. Link prediction on evolving data using matrix and tensor factorizations. In *2009 IEEE International Conference on Data Mining Workshops*, pages 262–269, Dec 2009.

[7] Lada A. Adamic and Eytan Adar. Friends and neighbors on the web. *Social Networks*, 25(3):211–230, 2003.

[8] Gediminas Adomavicius and Alexander Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Trans. on Knowl. and Data Eng.*, 17(6):734–749, June 2005.

[9] Apoorv Agarwal, Boyi Xie, Ilia Vovsha, Owen Rambow, and Rebecca Passonneau. Sentiment analysis of twitter data. In *Proceedings of the Workshop on Languages in Social Media*, LSM '11, pages 30–38, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics.

[10] Charu C Aggarwal. An introduction to social network data analytics. In *Social network data analytics*, pages 1–15. Springer, 2011.

[11] Luca Maria Aiello, Alain Barrat, Rossano Schifanella, Ciro Cattuto, Benjamin Markines, and Filippo Menczer. Friendship prediction and homophily in social media. *ACM Trans. Web*, 6(2):9:1–9:33, June 2012.

[12] Mohammad Al Hasan and Mohammed J Zaki. A survey of link prediction in social networks. In *Social network data analytics*, pages 243–275. Springer, 2011.

[13] Aris Anagnostopoulos, Ravi Kumar, and Mohammad Mahdian. Influence and correlation in social networks. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 7–15. ACM, 2008.

[14] Marcelo G. Armentano, Daniela Godoy, and Analía Amandi. Topology-based recommendation of users in micro-blogging communities. *Journal of Computer Science and Technology*, 27(3):624–634, 2012.

[15] Giuliano Arru, Davide Feltoni Gurini, Fabio Gasparetti, Alessandro Micarelli, and Giuseppe Sansonetti. Signal-based user recommendation on twitter. *Social Recommender Systems 2013*, 2013.

[16] Giuliano Arru, Davide Feltoni Gurini, Fabio Gasparetti, Alessandro Micarelli, and Giuseppe Sansonetti. Signal-based user recommendation on twitter. In *Proc. of the 22nd International Conference on World Wide Web Companion*, pages 941–944, 2013.

[17] Giuliano Arru, Davide Feltoni Gurini, Fabio Gasparetti, Alessandro Micarelli, and Giuseppe Sansonetti. Signal-based user recommendation on twitter. In *Proceedings of the 22nd International Conference on World Wide Web Companion*, pages 941–944, 2013.

[18] S Baccianella, A Esuli, and F Sebastiani. Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. 0: 2200-2204, 2008.

[19] Ricardo Baeza-Yates and Berthier Ribeiro-Neto. *Modern Information Retrieval*. Addison-Wesley Publishing Company, USA, 2nd edition, 2008.

[20] Marko Balabanović and Yoav Shoham. Fab: content-based, collaborative recommendation. *Communications of the ACM*, 40(3):66–72, 1997.

[21] Mathias Bank and Juergen Franke. Social networks as data source for recommendation systems. In Francesco Buccafurri and Giovanni Semeraro, editors, *E-Commerce and Web Technologies*, volume 61 of *Lecture Notes in Business Information Processing*, pages 49–60. Springer Berlin Heidelberg, 2010.

[22] Claudio Biancalana, Fabio Gasparetti, Alessandro Micarelli, Alfonso Miola, and Giuseppe Sansonetti. Context-aware movie recommendation based on signal processing and machine learning. In *Proceedings of the 2nd Challenge on Context-Aware Movie Recommendation*, CAMRa '11, pages 5–10, New York, NY, USA, 2011. ACM.

[23] Claudio Biancalana, Fabio Gasparetti, Alessandro Micarelli, Alfonso Miola, and Giuseppe Sansonetti. Wavelet-based music recommendation. In *WEBIST 2012 - Proceedings of the 8th International Conference on Web Information Systems and Technologies, Porto, Portugal, 18 - 21 April, 2012*, pages 399–402, 2012.

[24] Claudio Biancalana, Fabio Gasparetti, Alessandro Micarelli, and Giuseppe Sansonetti. Social semantic query expansion. *ACM Trans. Intell. Syst. Technol.*, 4(4), 2013. Forthcoming issue.

[25] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, March 2003.

[26] Johan Bollen, Huina Mao, and Alberto Pepe. Modeling public mood and emotion: Twitter sentiment and socio-economic phenomena. In *ICWSM'11*, pages –1–1, 2011.

[27] John S. Breese, David Heckerman, and Carl Kadie. Empirical analysis of predictive algorithms for collaborative filtering. 1998.

[28] Robin Burke. Hybrid recommender systems: Survey and experiments. *User modeling and user-adapted interaction*, 12(4):331–370, 2002.

[29] Sirian Caldarelli, Davide Feltoni Gurini, Alessandro Micarelli, and Giuseppe Sansonetti. A signal-based approach to news recommendation.

[30] Paolo Casoto, Antonina Dattolo, Paolo Omero, Nirmala Pudota, and Carlo Tasso. A new machine learning based approach for sentiment classification of italian documents. In *IRCDL*, pages 77–82, 2008.

[31] Òscar Celma and Pedro Cano. From hits to niches?: Or how popular artists can bias music recommendation and discovery. In *Proceedings of the 2Nd KDD Workshop on Large-Scale Recommender Systems and the Netflix Prize Competition*, NETFLIX '08, pages 5:1–5:8, New York, NY, USA, 2008. ACM.

[32] Sergiu Chelaru, Claudia Orellana-Rodriguez, and Ismail Sengor Altingovde. How useful is social feedback for learning to rank youtube videos? *World Wide Web*, 17(5):997–1025, 2014.

[33] Jilin Chen, Werner Geyer, Casey Dugan, Michael Muller, and Ido Guy. Make new friends, but keep the old: recommending people on social networking sites. In *Proceedings of the 27th International Conference on Human Factors in Computing Systems*, CHI '09, pages 201–210, New York, NY, USA, 2009. ACM.

[34] Jilin Chen, Werner Geyer, Casey Dugan, Michael Muller, and Ido Guy. Make new friends, but keep the old: Recommending people on social networking sites. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '09, pages 201–210, New York, NY, USA, 2009. ACM.

[35] Wen-Yen Chen, Jon-Chyuan Chu, Junyi Luan, Hongjie Bai, Yi Wang, and Edward Y Chang. Collaborative filtering for orkut communities: discovery of user latent behavior. In *Proceedings of the 18th international conference on World wide web*, pages 681–690. ACM, 2009.

[36] Joana Costa, Catarina Silva, Mário Antunes, and Bernardete Ribeiro. *Adaptive and Natural Computing Algorithms: 11th International Conference, ICANNGA 2013, Lausanne, Switzerland, April 4-6, 2013. Proceedings*, chapter Defining Semantic Meta-hashtags for Twitter Classification, pages 226–235. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.

[37] Isaac G. Councill, Ryan McDonald, and Leonid Velikovich. What's great and what's not: learning to classify the scope of negation for improved sentiment analysis. In *Proceedings of the Workshop on Negation and Speculation in Natural Language*

Processing, NeSp-NLP '10, pages 51–59, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

[38] Henriette Cramer, Vanessa Evers, Satyan Ramlal, Maarten Van Someren, Lloyd Rutledge, Natalia Stash, Lora Aroyo, and Bob Wielinga. The effects of transparency on trust in and acceptance of a content-based art recommender. *User Modeling and User-Adapted Interaction*, 18(5):455–496, 2008.

[39] Anqi Cui, Min Zhang, Yiqun Liu, Shaoping Ma, and Kuo Zhang. Discover breaking events with popular hashtags in twitter. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, CIKM '12, pages 1794–1798, New York, NY, USA, 2012. ACM.

[40] James Davidson, Benjamin Liebald, Junning Liu, Palash Nandy, Taylor Van Vleet, Ullas Gargi, Sujoy Gupta, Yu He, Mike Lambert, Blake Livingston, et al. The youtube video recommendation system. In *Proceedings of the fourth ACM conference on Recommender systems*, pages 293–296. ACM, 2010.

[41] Marco de Gemmis, Pasquale Lops, Giovanni Semeraro, and Cataldo Musto. An investigation on the serendipity problem in recommender systems. *Inf. Process. Manage.*, 51(5):695–717, September 2015.

[42] Dario De Nart and Carlo Tasso. A personalized concept-driven recommender system for scientific libraries. *Procedia Computer Science*, 38:84–91, 2014.

[43] Kerstin Denecke. Using sentiwordnet for multilingual sentiment analysis. In *Data Engineering Workshop, 2008. ICDEW 2008. IEEE 24th International Conference on*, pages 507–512. IEEE, 2008.

[44] Mukund Deshpande and George Karypis. Item-based top-n recommendation algorithms. *ACM Transactions on Information Systems (TOIS)*, 22(1):143–177, 2004.

[45] Christian Desrosiers and George Karypis. A comprehensive survey of neighborhood-based recommendation methods. In Francesco Ricci, Lior Rokach, Bracha Shapira, and Paul B. Kantor, editors, *Recommender Systems Handbook*, pages 107–144. Springer, 2011.

[46] Inderjit S Dhillon, Yuqiang Guan, and Brian Kulis. Weighted graph cuts without eigenvectors a multilevel approach. *IEEE transactions on pattern analysis and machine intelligence*, 29(11):1944–1957, 2007.

[47] Jordan Ellenberg. The netflix challenge. *WIRED-SAN FRANCISCO-*, 16(3):114, 2008.

[48] Andrea Esuli and Fabrizio Sebastiani. Pageranking wordnet synsets: An application to opinion mining. In *ACL*, volume 7, pages 442–431. Citeseer, 2007.

[49] Siamak Faridani. Using canonical correlation analysis for generalized sentiment analysis, product recommendation and search. In *Proceedings of the fifth ACM Conference on Recommender systems*, RecSys '11, pages 355–358, New York, NY, USA, 2011. ACM.

[50] Davide Feltoni Gurini, Fabio Gasparetti, Alessandro Micarelli, and Giuseppe Sansonetti. A sentiment-based approach to twitter user recommendation. In *RSWeb@RecSys*, 2013.

[51] Gary William Flake, Steve Lawrence, and C Lee Giles. Efficient identification of web communities. In *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 150–160. ACM, 2000.

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

[53] Eibe Frank and Remco R Bouckaert. Naive bayes for text classification with unbalanced classes. In *European Conference on Principles of Data Mining and Knowledge Discovery*, pages 503–510. Springer, 2006.

[54] Jill Freyne, Michal Jacovi, Ido Guy, and Werner Geyer. Increasing engagement through early recommender intervention. In *Proceedings of the third ACM Conference on Recommender Systems*, RecSys '09, pages 85–92, New York, NY, USA, 2009. ACM.

[55] Jill Freyne, Michal Jacovi, Ido Guy, and Werner Geyer. Increasing engagement through early recommender intervention. In *Proceedings of the third ACM Conference on Recommender Systems*, RecSys '09, pages 85–92, New York, NY, USA, 2009. ACM.

[56] Jill Freyne, Michal Jacovi, Ido Guy, and Werner Geyer. Increasing engagement through early recommender intervention. In *Proceedings of the Third ACM Conference on Recommender Systems*, RecSys '09, pages 85–92, New York, NY, USA, 2009. ACM.

[57] Michele Galli, Davide Feltoni Gurini, Fabio Gasparetti, Alessandro Micarelli, and Giuseppe Sansonetti. Analysis of user-generated content for improving youtube video recommendation. 2015.

[58] Michael Gamon, Anthony Aue, Simon Corston-Oliver, and Eric Ringger. Pulse: Mining customer opinions from free text. In *international symposium on intelligent data analysis*, pages 121–132. Springer, 2005.

[59] G Ganapathibhotla and Bing Liu. Identifying preferred entities in comparative sentences. In *Proceedings of the International Conference on Computational Linguistics, COLING*, 2008.

[60] Sheng Gao, Ludovic Denoyer, and Patrick Gallinari. Temporal link prediction by integrating content and structure information. In *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*, CIKM '11, pages 1169–1174, New York, NY, USA, 2011. ACM.

[61] Fabio Gasparetti, Carla Limongelli, and Filippo Sciarrone. A content-based approach for supporting teachers in discovering dependency relationships between instructional units in distance learning environments. In *International Conference on Human-Computer Interaction*, pages 241–246. Springer, 2015.

[62] Lise Getoor. Link mining and link discovery. In *Encyclopedia of Machine Learning*, pages 606–609. Springer, 2011.

[63] Werner Geyer, Casey Dugan, David R. Millen, Michael Muller, and Jill Freyne. Recommending topics for self-descriptions in online user profiles. In *Proceedings of the 2008 ACM Conference on Recommender Systems*, RecSys '08, pages 59–66, New York, NY, USA, 2008. ACM.

[64] M Ghiassi, J Skinner, and D Zimbra. Twitter brand sentiment analysis: A hybrid system using n-gram analysis and dynamic artificial neural network. *Expert Systems with applications*, 40(16):6266–6282, 2013.

[65] Alec Go, Richa Bhayani, and Lei Huang. Twitter sentiment classification using distant supervision. *Processing*, pages 1–6, 2009.

[66] Google. Google news. Last visited on 15 April 2016.

[67] M.S. Granovetter. The Strength of Weak Ties. *The American Journal of Sociology*, 78(6):1360–1380, 1973.

[68] Amara Graps. An introduction to wavelets. *IEEE Computational Science and Engineering*, 2(2), 1995.

[69] Georg Groh, Stefan Birnkammerer, and Valeria Köllhofer. *Recommender Systems for the Social Web*, chapter Social Recommender Systems, pages 3–42. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[70] Jingfeng Guo and Hongwei Guo. Multi-features link prediction based on matrix. In *2010 International Conference On Computer Design and Applications*, volume 1, pages V1–357–V1–361, June 2010.

[71] Davide Feltoni Gurini, Fabio Gasparetti, Alessandro Micarelli, and Giuseppe Sansonetti. iscur: Interest and sentiment-based community detection for user recommendation on twitter. In *International Conference on User Modeling, Adaptation, and Personalization*, pages 314–319. Springer International Publishing, 2014.

[72] Davide Feltoni Gurini, Fabio Gasparetti, Alessandro Micarelli, and Giuseppe Sansonetti. Analysis of sentiment communities in online networks. 2015.

[73] Davide Feltoni Gurini, Fabio Gasparetti, Alessandro Micarelli, and Giuseppe Sansonetti. Enhancing social recommendation with sentiment communities. In *International Conference on Web Information Systems Engineering*, pages 308–315. Springer International Publishing, 2015.

[74] Davide Feltoni Gurini, Fabio Gasparetti, Alessandro Micarelli, and Giuseppe Sansonetti. Temporal people-to-people recommendation on social networks with sentiment-based matrix factorization. *Future Generation Computer Systems*, pages – , 2017.

[75] Ido Guy, Michal Jacovi, Elad Shahar, Noga Meshulam, Vladimir Soroka, and Stephen Farrell. Harvesting with sonar: The value of aggregating social network information. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '08, pages 1017–1026, New York, NY, USA, 2008. ACM.

[76] Ido Guy, Inbal Ronen, and Eric Wilcox. Do you know?: recommending people to invite into your social network. In *Proceedings of the 14th International Conference on Intelligent User Interfaces*, IUI '09, pages 77–86, New York, NY, USA, 2009. ACM.

[77] Ido Guy, Inbal Ronen, and Eric Wilcox. Do you know?: Recommending people to invite into your social network. In *Proceedings of the 14th International Conference on Intelligent User Interfaces*, IUI '09, pages 77–86, New York, NY, USA, 2009. ACM.

[78] Ido Guy, Naama Zwerdling, David Carmel, Inbal Ronen, Erel Uziel, Sivan Yogev, and Shila Ofek-Koifman. Personalized recommendation of social software items based on social relations. In *Proceedings of the third ACM conference on Recommender systems*, pages 53–60. ACM, 2009.

[79] Ido Guy, Naama Zwerdling, Inbal Ronen, David Carmel, and Erel Uziel. Social media recommendation based on people and tags. In *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '10, pages 194–201, New York, NY, USA, 2010. ACM.

[80] John Hannon, Mike Bennett, and Barry Smyth. Recommending twitter users to follow using content and collaborative filtering approaches. In *Proceedings of the fourth ACM Conference on Recommender Systems*, RecSys '10, pages 199–206. ACM, 2010.

[81] John Hannon, Mike Bennett, and Barry Smyth. Recommending twitter users to follow using content and collaborative filtering approaches. *RecSys'10 : Proceedings of the 4th ACM Conference on Recommender Systems,*, 26-30(10):8, 09 2010.

[82] John Hannon, Mike Bennett, and Barry Smyth. Recommending twitter users to follow using content and collaborative filtering approaches. In *Proceedings of the Fourth ACM Conference on Recommender Systems*, RecSys '10, pages 199–206, New York, NY, USA, 2010. ACM.

[83] John Hannon, Kevin McCarthy, Michael P. O'Mahony, and Barry Smyth. A multi-faceted user model for twitter. In *Proceedings of the 20th International Conference on User Modeling, Adaptation, and Personalization*, UMAP'12, pages 303–309, 2012.

[84] John Hannon, Kevin McCarthy, and Barry Smyth. Finding useful users on twitter: twittomender the followee recommender. In *Proceedings of the 33rd European Conference on Advances in Information Retrieval*, ECIR'11, pages 784–787, Berlin, Heidelberg, 2011. Springer-Verlag.

[85] Mohammad Al Hasan, Vineet Chaoji, Saeed Salem, and Mohammed Zaki. Link prediction using supervised learning. In *In Proc. of SDM 06 workshop on Link Analysis, Counterterrorism and Security*, 2006.

[86] Jonathan L. Herlocker, Joseph A. Konstan, Loren G. Terveen, and John T. Riedl. Evaluating collaborative filtering recommender systems. *ACM Trans. Inf. Syst.*, 22(1):5–53, January 2004.

[87] Yoshinori Hijikata, Takuya Shimizu, and Shogo Nishida. Discovery-oriented collaborative filtering for improving user satisfaction. In *Proceedings of the 14th International Conference on Intelligent User Interfaces*, IUI '09, pages 67–76, New York, NY, USA, 2009. ACM.

[88] Minqing Hu and Bing Liu. Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 168–177. ACM, 2004.

[89] Xia Hu, Jiliang Tang, Huiji Gao, and Huan Liu. Unsupervised sentiment analysis with emotional signals. In *Proceedings of the 22nd international conference on World Wide Web*, pages 607–618. ACM, 2013.

[90] Yifan Hu, Yehuda Koren, and Chris Volinsky. Collaborative filtering for implicit feedback datasets. In *Proceedings of the 2008 Eighth IEEE International Conference on Data Mining*, ICDM '08, pages 263–272, Washington, DC, USA, 2008. IEEE Computer Society.

[91] Zan Huang, Daniel Zeng, and Hsinchun Chen. A comparison of collaborative-filtering recommendation algorithms for e-commerce. *IEEE Intelligent Systems*, 22(5):68–78, 2007.

[92] Ashutosh Sopan Jadhav, Hemant Purohit, Pavan Kapanipathi, Pramod Anantharam, Ajith H Ranabahu, Vinh Nguyen, Pablo N Mendes, Alan Gary Smith, Michael Cooney, and Amit P Sheth. Twitris 2.0: Semantically empowered system for understanding perceptions from social data. 2010.

[93] Thorsten Joachims. Text categorization with suport vector machines: Learning with many relevant features. In *Proceedings of the 10th European Conference on Machine Learning*, ECML '98, pages 137–142, London, UK, UK, 1998. Springer-Verlag.

[94] Thorsten Joachims. Text categorization with support vector machines: Learning with many relevant features. In *European conference on machine learning*, pages 137–142. Springer, 1998.

[95] Thorsten Joachims, Laura Granka, Bing Pan, Helene Hembrooke, and Geri Gay. Accurately interpreting clickthrough data as implicit feedback. In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '05, pages 154–161, New York, NY, USA, 2005. ACM.

[96] Hannon John, Bennett Mike, and Smyth Barry. Recommending twitter users to follow using content and collaborative filtering approaches. *RecSys'10 : Proceedings of the 4th ACM Conference on Recommender Systems,*, 26-30(10):8, 09 2010.

[97] Andreas Kanavos and Isidoros Perikos. Towards detecting emotional communities in twitter. In *2015 IEEE 9th International Conference on Research Challenges in Information Science (RCIS)*, pages 524–525. IEEE, 2015.

[98] Hiroshi Kanayama and Tetsuya Nasukawa. Fully automatic lexicon expansion for domain-oriented sentiment analysis. In *Proceedings of the 2006 conference on empirical methods in natural language processing*, pages 355–363. Association for Computational Linguistics, 2006.

[99] Alexandros Karatzoglou, Xavier Amatriain, Linas Baltrunas, and Nuria Oliver. Multiverse recommendation: N-dimensional tensor factorization for context-aware collaborative filtering. In *Proceedings of the Fourth ACM Conference on Recommender Systems*, RecSys '10, pages 79–86, New York, NY, USA, 2010. ACM.

[100] Danae Pla Karidi. From user graph to topics graph: Towards twitter followee recommendation based on knowledge graphs. In *Data Engineering Workshops (ICDEW), 2016 IEEE 32nd International Conference on*, pages 121–123. IEEE, 2016.

[101] George Karypis and Vipin Kumar. A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM Journal on scientific Computing*, 20(1):359–392, 1998.

[102] David Kempe, Jon Kleinberg, and Éva Tardos. Maximizing the spread of influence through a social network. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 137–146. ACM, 2003.

[103] Brian W Kernighan and Shen Lin. An efficient heuristic procedure for partitioning graphs. *Bell system technical journal*, 49(2):291–307, 1970.

[104] Kyoung-jae Kim and Hyunchul Ahn. A recommender system using ga k-means clustering in an online shopping market. *Expert systems with applications*, 34(2):1200–1209, 2008.

[105] Yehuda Koren. Factorization meets the neighborhood: A multifaceted collaborative filtering model. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '08, pages 426–434, New York, NY, USA, 2008. ACM.

[106] Yehuda Koren. Collaborative filtering with temporal dynamics. *Communications of the ACM*, 53(4):89–97, 2010.

[107] Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, August 2009.

[108] Onur Kucuktunc, B Barla Cambazoglu, Ingmar Weber, and Hakan Ferhatosmanoglu. A large-scale sentiment analysis for yahoo! answers. In *Proceedings of the fifth ACM international conference on Web search and data mining*, pages 633–642. ACM, 2012.

[109] Sangeetha Kutty, Lin Chen, and Richi Nayak. A people-to-people recommendation system using tensor space models. In *Proceedings of the 27th Annual ACM Symposium on Applied Computing*, SAC '12, pages 187–192, New York, NY, USA, 2012. ACM.

[110] Thomas Lake and William Fitzgerald. Twitter sentiment analysis. *Western Michigan University, Kalamazoo, MI*, 2011.

[111] Lieven De Lathauwer, Bart De Moor, and Joos Vandewalle. A multilinear singular value decomposition. *SIAM J. Matrix Anal. Appl.*, 21(4):1253–1278, March 2000.

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

[113] Jure Leskovec, Jon Kleinberg, and Christos Faloutsos. Graphs over time: densification laws, shrinking diameters and possible explanations. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pages 177–187. ACM, 2005.

[114] Mark R. Levy and S. V. E. N. Windahl. AUDIENCE ACTIVITY AND GRATIFICATIONS: A Conceptual Clarification and Exploration. *Communication Research*, 11(1):51–78, January 1984.

[115] David Liben-Nowell and Jon Kleinberg. The link prediction problem for social networks. In *Proceedings of the Twelfth International Conference on Information and Knowledge Management*, CIKM '03, pages 556–559, New York, NY, USA, 2003. ACM.

[116] Bing Liu. Sentiment analysis and subjectivity. *Handbook of Natural Language Processing,*, pages 627–666, 2010.

[117] Jiahui Liu, Elin Pedersen, and Peter Dolan. Personalized news recommendation based on click behavior. In *2010 International Conference on Intelligent User Interfaces*, 2010.

[118] Pasquale Lops, Marco de Gemmis, and Giovanni Semeraro. Content-based recommender systems: State of the art and trends. In Francesco Ricci, Lior Rokach, Bracha Shapira, and Paul B. Kantor, editors, *Recommender Systems Handbook*, pages 73–105. Springer, 2011.

[119] Hao Ma, Haixuan Yang, Michael R Lyu, and Irwin King. Sorec: social recommendation using probabilistic matrix factorization. In *Proceedings of the 17th ACM conference on Information and knowledge management*, pages 931–940. ACM, 2008.

[120] S. G. Mallat. A theory for multiresolution signal decomposition: The wavelet representation. *IEEE Trans. on PAMI*, 11(7):674–693, July 1989.

[121] Nimita Mangal, Rajdeep Niyogi, and Alfredo Milani. Analysis of users' interest based on tweets. In *International Conference on Computational Science and Its Applications*, pages 12–23. Springer, 2016.

[122] PV Marsden and NE Friedkin. Network studies of social influence sociological methods & research, 1993 sage publications. 1993.

[123] S. M. McNee, J. Riedl, and J.A. Konstan. Accurate is not always good: How accuracy metrics have hurt recommender systems. In *Extended Abstracts of the 2006 ACM Conference on Human Factors in Computing Systems (CHI 2006)*, 2006.

[124] Miller McPherson, Lynn S. Lovin, and James M. Cook. Birds of a Feather: Homophily in Social Networks. *Annual Review of Sociology*, 27(1):415–444, 2001.

[125] Aditya Krishna Menon and Charles Elkan. Link prediction via matrix factorization. In *Proceedings of the 2011 European Conference on Machine Learning and Knowledge Discovery in Databases - Volume Part II*, ECML PKDD'11, pages 437–452, Berlin, Heidelberg, 2011. Springer-Verlag.

[126] Andrew Messenger and Jon Whittle. Recommendations based on user-generated comments in social media. In *Privacy, Security, Risk and Trust (PASSAT) and 2011 IEEE Third Inernational Conference on Social Computing (SocialCom), 2011 IEEE Third International Conference on*, pages 505–508. IEEE, 2011.

[127] Stuart E Middleton, David C De Roure, and Nigel R Shadbolt. Capturing knowledge of user preferences: ontologies in recommender systems. In *Proceedings of the 1st international conference on Knowledge capture*, pages 100–107. ACM, 2001.

[128] Stanley Milgram. The Small World Problem. *Psychology Today*, 2:60–67, 1967.

[129] George Miller and Christiane Fellbaum. Wordnet: An electronic lexical database, 1998.

[130] Tomoko Murakami, Koichiro Mori, and Ryohei Orihara. Metrics for evaluating the serendipity of recommendation lists. In *Proceedings of the 2007 Conference on New Frontiers in Artificial Intelligence*, JSAI'07, pages 40–46, Berlin, Heidelberg, 2008. Springer-Verlag.

[131] Mark EJ Newman and Michelle Girvan. Finding and evaluating community structure in networks. *Physical review E*, 69(2):026113, 2004.

[132] Thin Nguyen, Dinh Q. Phung, Brett Adams, and Svetha Venkatesh. A sentiment-aware approach to community formation in social media. In John G. Breslin, Nicole B. Ellison, James G. Shanahan, and Zeynep Tufekci, editors, *ICWSM*. The AAAI Press, 2012.

[133] Maria Augusta Silveira Netto Nunes and Stefano A Cerri. Recommender systems based on human psychological reputation. In *Recommenders' 06. com-THE PRESENT AND FUTURE OF RECOMMENDER SYSTEMS*, pages http–blog. MyStrands, 2006.

[134] Edward Rolando Núñez-Valdéz, Juan Manuel Cueva Lovelle, Oscar Sanjuán Martínez, Vicente García-Díaz, Patricia Ordoñez de Pablos, and Carlos Enrique Montenegro Marín. Implicit feedback techniques on recommender systems applied to electronic books. *Computers in Human Behavior*, 28(4):1186–1193, 2012.

[135] Alexander Pak and Patrick Paroubek. Twitter as a corpus for sentiment analysis and opinion mining. volume 2010, pages 1320–1326, 2010.

[136] Alexander Pak and Patrick Paroubek. Twitter as a corpus for sentiment analysis and opinion mining. volume 2010, pages 1320–1326, 2010.

[137] Alexander Pak and Patrick Paroubek. Twitter as a corpus for sentiment analysis and opinion mining. In Nicoletta Calzolari, Khalid Choukri, Bente Maegaard, Joseph Mariani, Jan Odijk, Stelios Piperidis, Mike Rosner, and Daniel Tapias, editors, *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta, May 2010. European Language Resources Association (ELRA).

[138] Bo Pang and Lillian Lee. Opinion mining and sentiment analysis. *Foundations and trends in information retrieval*, 2(1-2):1–135, 2008.

[139] Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. Thumbs up?: sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*, pages 79–86. Association for Computational Linguistics, 2002.

[140] Laurence A. F. Park, Kotagiri Ramamohanarao, and Marimuthu Palaniswami. A novel document retrieval method using the discrete wavelet transform. *ACM Trans. Inf. Syst.*, 23(3):267–298, July 2005.

[141] Thi-Ngan Pham, Thi-Hong Vuong, Thi-Hoai Thai, Mai-Vu Tran, and Quang-Thuy Ha. Sentiment analysis and user similarity for social recommender system: An experimental study. In *Information Science and Applications (ICISA) 2016*, pages 1147–1156. Springer, 2016.

[142] Owen Phelan, Kevin McCarthy, and Barry Smyth. Using twitter to recommend real-time topical news. In *Proceedings of the third ACM conference on Recommender systems*, pages 385–388. ACM, 2009.

[143] Carlos Porcel and Enrique Herrera-Viedma. Dealing with incomplete information in a fuzzy linguistic recommender system to disseminate information in university digital libraries. *Knowledge-Based Systems*, 23(1):32–39, 2010.

[144] Carlos Porcel, Juan Manuel Moreno, and Enrique Herrera-Viedma. A multi-disciplinar recommender system to advice research resources in university digital libraries. *Expert Systems with Applications*, 36(10):12520–12528, 2009.

[145] Rudy Prabowo and Mike Thelwall. Sentiment analysis: A combined approach. *Journal of Informetrics*, 3(2):143–157, 2009.

[146] Daniele Quercia and Licia Capra. Friendsensing: Recommending friends using mobile phones. In *Proceedings of the Third ACM Conference on Recommender Systems*, RecSys '09, pages 273–276, New York, NY, USA, 2009. ACM.

[147] Benjamin Renoust, Guy Melançon, and Marie-Luce Viaud. Entanglement in multiplex networks: understanding group cohesion in homophily networks. In *Social Network Analysis-Community Detection and Evolution*, pages 89–117. Springer, 2014.

[148] Paul Resnick, Neophytos Iacovou, Mitesh Suchak, Peter Bergstrom, and John Riedl. Grouplens: an open architecture for collaborative filtering of netnews. In *Proceedings of the 1994 ACM conference on Computer supported cooperative work*, pages 175–186. ACM, 1994.

[149] Francesco Ricci, Lior Rokach, and Bracha Shapira. *Introduction to recommender systems handbook*. Springer, 2011.

[150] Takeshi Sakaki, Makoto Okazaki, and Yutaka Matsuo. Earthquake shakes twitter users: real-time event detection by social sensors. In Michael Rappa, Paul Jones, Juliana Freire, and Soumen Chakrabarti, editors, *WWW*, pages 851–860. ACM, 2010.

[151] Jagan Sankaranarayanan, Hanan Samet, Benjamin E. Teitler, Michael D. Lieberman, and Jon Sperling. Twitterstand: News in tweets. In *Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, GIS '09, pages 42–51, New York, NY, USA, 2009. ACM.

[152] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th International Conference on World Wide Web*, WWW '01, pages 285–295, New York, NY, USA, 2001. ACM.

[153] Venu Satuluri and Srinivasan Parthasarathy. Scalable graph clustering using stochastic flows: applications to community discovery. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 737–746. ACM, 2009.

[154] Andrew I Schein, Alexandrin Popescul, Lyle H Ungar, and David M Pennock. Methods and metrics for cold-start recommendations. In *Proceedings of the 25th annual*

international ACM SIGIR conference on Research and development in information retrieval, pages 253–260. ACM, 2002.

[155] John Scott. *Social network analysis*. Sage, 2012.

[156] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Transactions on pattern analysis and machine intelligence*, 22(8):888–905, 2000.

[157] Erez Shmueli, Amit Kagian, Yehuda Koren, and Ronny Lempel. Care to comment?: recommendations for commenting on news stories. In *Proceedings of the 21st international conference on World Wide Web*, pages 429–438. ACM, 2012.

[158] Börkur Sigurbjörnsson and Roelof Van Zwol. Flickr tag recommendation based on collective knowledge. In *Proceedings of the 17th international conference on World Wide Web*, pages 327–336. ACM, 2008.

[159] Vivek Kumar Singh, Mousumi Mukherjee, and Ghanshyam Kumar Mehta. Combining collaborative filtering and sentiment classification for improved movie recommendations. In *Proceedings of the 5th International Conference on Multi-Disciplinary Trends in Artificial Intelligence*, MIWAI'11, pages 38–50, Berlin, Heidelberg, 2011. Springer-Verlag.

[160] Daniel A Spielman and Nikhil Srivastava. Graph sparsification by effective resistances. *SIAM Journal on Computing*, 40(6):1913–1926, 2011.

[161] Panagiotis Symeonidis, Alexandros Nanopoulos, and Yannis Manolopoulos. Providing justifications in recommender systems. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 38(6):1262–1272, 2008.

[162] Seyed Amin Tabatabaei and Masoud Asadpour. Study of influential trends, communities, and websites on the post-election events of iranian presidential election in twitter. In *Social Network Analysis-Community Detection and Evolution*, pages 71–87. Springer, 2014.

[163] Taffee T. Tanimoto. An elementary mathematical theory of classification and prediction. *IBM Internal Report*, 1957.

[164] Mike Thelwall. Emotion homophily in social network site messages. *First Monday*, 15(4), 2010.

[165] Mike Thelwall, Kevan Buckley, and Georgios Paltoglou. Sentiment in twitter events. *J. Am. Soc. Inf. Sci. Technol.*, 62(2):406–418, February 2011.

[166] Roberto Torres, Sean M McNee, Mara Abel, Joseph A Konstan, and John Riedl. Enhancing digital libraries with techlens+. In *Proceedings of the 4th ACM/IEEE-CS joint conference on Digital libraries*, pages 228–236. ACM, 2004.

[167] A. Tumasjan, T.O. Sprenger, P.G. Sandner, and I.M. Welpe. Predicting elections with twitter: What 140 characters reveal about political sentiment. In *Proceedings of the Fourth International AAAI Conference on Weblogs and Social Media*, pages 178–185, 2010.

[168] Peter D Turney. Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 417–424. Association for Computational Linguistics, 2002.

[169] Twitter Inc. About twitter's account suggestions. Last visited on 20 December 2016.

[170] Yulia Tyshchuk, Hao Li, Heng Ji, and William A Wallace. The emergence of communities and their leaders on twitter following an extreme event. In *Social Network Analysis-Community Detection and Evolution*, pages 1–25. Springer, 2014.

[171] Stijn Marinus Van Dongen. Graph clustering by flow simulation. 2001.

[172] Ulrike Von Luxburg. A tutorial on spectral clustering. *Statistics and computing*, 17(4):395–416, 2007.

[173] Peng Wang, BaoWen Xu, YuRong Wu, and XiaoYu Zhou. Link prediction in social networks: the state-of-the-art. *Science China Information Sciences*, 58(1):1–38, 2015.

[174] Xi Wang and Gita Sukthankar. Link prediction in heterogeneous collaboration networks. In *Social network analysis-community detection and evolution*, pages 165–192. Springer, 2014.

[175] Roy Want. An introduction to rfid technology. *IEEE Pervasive Computing*, 5(1):25–33, 2006.

[176] Stanley Wasserman and Katherine Faust. *Social network analysis: Methods and applications*, volume 8. Cambridge university press, 1994.

[177] Duncan J. Watts and Steven H. Strogatz. Collective dynamics of small-world networks. *Nature*, 393(6684):440–442, June 1998.

[178] Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of the conference on human language technology and empirical methods in natural language processing*, pages 347–354. Association for Computational Linguistics, 2005.

[179] Manda Winlaw, Michael B. Hynes, Anthony Caterini, and Hans De Sterck. Algorithmic acceleration of parallel ALS for collaborative filtering: Speeding up distributed big data recommendation in spark. In *21st IEEE International Conference on Parallel and Distributed Systems, ICPADS 2015, Melbourne, Australia, December 14-17, 2015*, pages 682–691, 2015.

[180] Xindong Wu, Vipin Kumar, J Ross Quinlan, Joydeep Ghosh, Qiang Yang, Hiroshi Motoda, Geoffrey J McLachlan, Angus Ng, Bing Liu, S Yu Philip, et al. Top 10 algorithms in data mining. *Knowledge and information systems*, 14(1):1–37, 2008.

[181] Peng Xia, Kun Tu, Bruno Ribeiro, Hua Jiang, Xiaodong Wang, Cindy Chen, Benyuan Liu, and Don Towsley. Characterization of user online dating behavior and preference on a large online dating site. In *Social Network Analysis-Community Detection and Evolution*, pages 193–217. Springer, 2014.

[182] Feng Xie, Ming Xu, and Zhen Chen. Rbra: A simple and efficient rating-based recommender algorithm to cope with sparsity in recommender systems. In Leonard Barolli, Tomoya Enokido, Fatos Xhafa, and Makoto Takizawa, editors, *AINA Workshops*, pages 306–311. IEEE Computer Society, 2012.

[183] Kaiquan Xu, Jiexun Li, and Stephen Shaoyi Liao. Sentiment community detection in social networks. In *Proc. of the 2011 iConference*, pages 804–805, New York, NY, USA, 2011. ACM.

[184] Yuki Yamamoto, Tadahiko Kumamoto, and Akiyo Nadamoto. Followee recommendation based on topic extraction and sentiment analysis from tweets. In *Proceedings of the 17th International Conference on Information Integration and Web-based Applications & Services*, page 27. ACM, 2015.

[185] Zhenlei Yan and Jie Zhou. User recommendation with tensor factorization in social networks. In *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*, pages 3853 –3856, March 2012.

[186] Dingqi Yang, Daqing Zhang, Zhiyong Yu, and Zhu Wang. A sentiment-enhanced personalized location recommendation system. In *Proceedings of the 24th ACM Conference on Hypertext and Social Media*, HT '13, pages 119–128, New York, NY, USA, 2013. ACM.

[187] Dingqi Yang, Daqing Zhang, Zhiyong Yu, and Zhiwen Yu. Fine-grained preference-aware location search leveraging crowdsourced digital footprints from lbsns. In *Proceedings of the 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, UbiComp '13, pages 479–488, New York, NY, USA, 2013. ACM.

[188] Guangchao Yuan, Pradeep K. Murukannaiah, Zhe Zhang, and Munindar P. Singh. Exploiting sentiment homophily for link prediction. In *Proceedings of the 8th ACM Conference on Recommender Systems*, RecSys '14, pages 17–24, New York, NY, USA, 2014. ACM.

[189] Guangchao Yuan, Pradeep K Murukannaiah, Zhe Zhang, and Munindar P Singh. Exploiting sentiment homophily for link prediction. In *Proceedings of the 8th ACM Conference on Recommender systems*, pages 17–24. ACM, 2014.

[190] Matei Zaharia, Mosharaf Chowdhury, Tathagata Das, Ankur Dave, Justin Ma, Murphy McCauley, Michael J. Franklin, Scott Shenker, and Ion Stoica. Resilient distributed datasets: A fault-tolerant abstraction for in-memory cluster computing. In *Proceedings of the 9th USENIX Conference on Networked Systems Design and Implementation*, NSDI'12, pages 2–2, Berkeley, CA, USA, 2012. USENIX Association.

[191] Jiawei Zhang, Yuanhua Lv, and Philip Yu. Enterprise social link recommendation. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, CIKM '15, pages 841–850, New York, NY, USA, 2015. ACM.

[192] Tong Zhang. Solving large scale linear prediction problems using stochastic gradient descent algorithms. In *Proceedings of the twenty-first international conference on Machine learning*, page 116. ACM, 2004.

[193] Yang Zhang, Yao Wu, and Qing Yang. Community discovery in twitter based on user interests. *Journal of Computational Information Systems*, 8(3):991–1000, 2012.

[194] Gang Zhao, Mong Li Lee, Wynne Hsu, Wei Chen, and Haoji Hu. Community-based user recommendation in uni-directional social networks. In *Proceedings of the 22Nd ACM International Conference on Information & Knowledge Management*, CIKM '13, pages 189–198, New York, NY, USA, 2013. ACM.

[195] Yunhong Zhou, Dennis Wilkinson, Robert Schreiber, and Rong Pan. Large-scale parallel collaborative filtering for the netflix prize. In *Proceedings of the 4th International Conference on Algorithmic Aspects in Information and Management*, AAIM '08, pages 337–348, Berlin, Heidelberg, 2008. Springer-Verlag.

[196] Yunhong Zhou, Dennis Wilkinson, Robert Schreiber, and Rong Pan. Large-scale parallel collaborative filtering for the netflix prize. In *Proceedings of the 4th International Conference on Algorithmic Aspects in Information and Management*, AAIM '08, pages 337–348, Berlin, Heidelberg, 2008. Springer-Verlag.

[197] Pamela Zontone, Giulia Boato, Jonathon Hare, Paul Lewis, Stefan Siersdorfer, and Enrico Minack. Image and collateral text in support of auto-annotation and sentiment analysis. In *Proceedings of the 2010 Workshop on Graph-based Methods for Natural Language Processing*, pages 88–92. Association for Computational Linguistics, 2010.