Social Network Analysis and Mining (2019) 9:64
https://doi.org/10.1007/s13278-019-0610-x

ORIGINAL ARTICLE

Handling uncertainty in social media textual information for improving venue recommendation formulation quality in social networks

Dionisis Margaris[1] · Costas Vassilakis[2] · Dimitris Spiliotopoulos[2]

Received: 18 April 2019 / Revised: 13 October 2019 / Accepted: 17 October 2019 / Published online: 31 October 2019
© Springer-Verlag GmbH Austria, part of Springer Nature 2019

Abstract

One of the major problems that social media front is to continuously produce successful, user-targeted information, in the form of recommendations, which are produced by applying methods from the area of recommender systems. One of the most important applications of recommender systems in social networks is venue recommendation, targeted by the majority of the leading social networks (Facebook, TripAdvisor, OpenTable, etc.). However, recommender systems' algorithms rely only on the existence of numeric ratings which are typically entered by users, and in the context of social networks, this information is scarce, since many social networks allow only reviews, rather than explicit ratings. Even if explicit ratings are supported, users may still resort to expressing their views and rating their experiences through submitting posts, which is the predominant user practice in social networks, rather than entering explicit ratings. User posts contain textual information, which can be exploited to compute derived ratings, and these derived ratings can be used in the recommendation process in the lack of explicitly entered ratings. Emerging recommender systems encompass this approach, without however tackling the fact that the ratings computed on the basis of textual information may be inaccurate, due to the very nature of the computation process. In this paper, we present an approach which extracts features of the textual information, a widely available source of information in venue category, to compute a confidence metric for the ratings that are computed from texts; then, this confidence metric is used in the user similarity computation and venue rating prediction formulation process, along with the computed rating. Furthermore, we propose a venue recommendation method that considers the generated venue rating predictions, along with venue QoS, similarity and spatial distance metrics in order to generate venue recommendations for social network users. Finally, we validate the accuracy of the rating prediction method and the user satisfaction from the recommendations generated by the recommendation formulation algorithm. Conclusively, the introduction of the confidence level significantly improves rating prediction accuracy, leverages the ability to generate personalized recommendations for users and increases user satisfaction.

Keywords Social networks · Recommender systems · Collaborative filtering · Venue recommendation formulation · Rating prediction · Textual information · Uncertainty · Confidence level

1 Introduction

With the advent of social networks (SNs), such as Facebook and Twitter, used by millions of people every day, large volumes of data generated by these networks are widely available, and both researchers and industry seek methods to exploit these data for personalization and recommendation purposes (Eirinaki et al. 2018; Margaris and Vassilakis 2018b; Bakshy et al. 2012a). These data are deemed of high value in the context of personalization because of the importance and the intrinsic relationship with people's everyday lives. Data are entered by users in various forms: posts on

✉ Costas Vassilakis
costas@uop.gr

[1] Department of Informatics and Telecommunications, University of Athens, Athens, Greece

[2] Department of Informatics and Telecommunications, University of the Peloponnese, Tripoli, Greece

their walls, or on the walls of others, reviews with explicit ratings, check-ins, likes, tagged photos and so forth. Many of the users' activities signify presence in venues (through the association with a location or through posts either on the page of a venue or on the page of an event that is held at the venue), while at the same time they include textual descriptions, in which users comment on their experiences, their mood and current status, and therefore conveying the attitude of users toward the associated venues.

Nowadays, different types of recommenders exist, which predominantly follow the collaborative filtering (CF) approach (Balabanovic and Shoham 1997; Tiago et al. 2018), and many SNs employ such algorithms to compute venue recommendations (as well as other recommendation categories) and deliver them to their users. Contemporary works in this area go beyond the examination of the item ratings entered by the users and consider additional aspects of item and user context, including metrics such as influence between users that participate in SNs, the semantic similarity between items and the items' qualitative characteristics, such as price and perceived quality. Typical recommender systems (RSs) rely on the existence of numeric ratings explicitly entered by the users, a type of information barely available in venue category; therefore, they are mostly limited to extracting and processing reviews, which include explicit numeric-scaled ratings entered by the users. This limitation however significantly reduces the potential of SN-based RSs to generate personalized recommendations for their users, since reviews constitute only a small fraction of the data that users contribute to SNs.

In the venue category, in particular, this phenomenon is extremely common. For instance, the Denver Art Museum has gathered 3990 ratings from Facebook users,[1] but the number of users who have shared a check-in in the same venue is 169,000.[2] Many posts and check-ins[3] are complete with a piece of text where the user describes her experience in the venue, and the post or check-in may be followed by a conversation between the user and her friends It is clear that this textual data hosted in the SN is a potential source of valuable information which can significantly enhance the coverage of the RS (i.e., its capability to produce personalized recommendations for its users), as well as its rating prediction and recommendation quality. This potential has been lately recognized, and a number of research efforts extend RSs with the ability to use textual information (Cocarascu and Toni 2018; Contratres et al. 2018; Maurya et al. 2018; Pero and Horváth 2013; Poirier et al. 2010; Moshfeghi et al. 2010; Raghavan et al. 2012) or post and check-in data (e.g., Margaris et al. 2017) to infer user ratings for venues, where explicitly entered ratings are not available.

However, the very nature of computing a rating based on textual data inherently introduces some uncertainty, due to the fuzziness of human language, and this aspect is not addressed in current research: All computed ratings are treated equivalently to explicitly entered ratings, despite the fact that explicitly entered ratings are known to be accurate, contrary to the case of computed ratings where errors may occur (Contratres et al. 2018).

In this paper, we present an approach which extracts features of the textual information within the posts on social media that are associated with venues, to compute a confidence metric for the ratings that are computed from texts. According to our approach, the textual data are extracted and processed to compute a numerical user rating, which is then tagged with a confidence metric. Different approaches for computing confidence are explored in this context. Then, the computed ratings and the associated confidence metrics are used both in the phases of nearest neighbor computation and venue rating prediction, in order to produce more reliable venue recommendations. Subsequently, we present a venue recommendation algorithm that combines venue rating predictions with metrics about the quality of service (QoS) offered by the venues, as well as the similarity and physical distance between the venues to generate personalized recommendations highly tailored to each user's personal profile. Finally, we validate the effectiveness of the proposed approach for rating prediction and for venue recommendation generation using both ground truth data and a user survey, and the results appear promising.

The rest of the paper is structured as follows: Sect. 2 overviews related work, while Sect. 3 presents the algorithm for computing (user rating, rating confidence) pairs based on textual data sourced from the SN, and a venue rating prediction algorithm which exploits computed ratings and associated confidences, together with user influence data; Sect. 3 also briefly introduces prerequisites for these operations. Section 4 describes the venue recommendation formulation algorithm for SNs, while Sect. 5 evaluates the proposed algorithms. Finally, Sect. 6 concludes the paper and outlines future work.

2 Related work

Collaborative filtering formulates personalized recommendations on the basis of ratings expressed by people having similar tastes to the user for whom the recommendation is

[1] https://www.facebook.com/pg/denverartmuseum/reviews/.

[2] https://www.facebook.com/pg/denverartmuseum/community/.

[3] The Facebook check-in post type has been deprecated in favor of creating a post with a location attached (https://developers.facebook.com/docs/graph-api/reference/v2.11/checkin); in the rest of the paper we will however use the term *check-in* to actually refer to these posts, which signify user presence at some place.

generated; taste similarity is computed by examining the resemblance of already entered ratings (Herlocker et al. 2004; Tiago et al. 2018); the CF-based recommendation approach is the most successful and widely used approach for implementing RSs (Tiago et al. 2018, Margaris and Vassilakis 2018c). To improve recommendation accuracy, knowledge-based RSs nowadays employ cutting-edge techniques such as data mining and segmentation (Yun et al. 2018).

RSs may be designed for multiple domains of application. RESYGEN (Monfil-Contreras et al. 2013) is a recommendation system generator that can generate multi-domain systems. For similarity computation in the recommendation process, RESYGEN provides a similarity metrics library and the RS configurator chooses the most appropriate one.

With the advent of SN, SN and RSs have received considerable research attention. Bakshy et al. (2012a) examine the role of SN in the recommendation process within a field experiment that randomizes exposure to signals about friends' information and the relative role of strong and weak ties. Bakshy et al. (2012b) measure social influence via social cues, demonstrate the substantial consequences of including minimal social cues in advertising and measure the positive relationship between a consumer's response and the strength of her connection with an affiliated peer. Both these works establish that recommendation algorithms are valuable tools in SN, and examine social cues and other methods to increase the probability that a recommendation is adopted.

In the domain of venue recommendation, textual information from check-ins is exploited in (Jameel et al. 2016); this work proposes a text mining framework for analyzing the lifestyles of users of location-based SNs such as Foursquare. In particular, the framework is based on a novel topic modeling approach, in which it explicitly addresses the sparsity of check-in data and incorporate a temporal component for analyzing how lifestyle patterns change throughout the year. The output of the method consists of a set of lifestyle patterns, each of which corresponds to a probability distribution over lifestyle topics, the latter intuitively corresponding to soft clusters of related venues. The computed lifestyle patterns can be later used as an input to RSs, for generating temporally aware recommendations.

Chen et al. (2015) provide a comprehensive review regarding the creation of ratings based on textual information. In that work, we can identify two main directions for textual review exploitation: the first direction is toward using textual reviews to derive ratings, when explicit ratings are not available. In this context, Fan and Khademi (2014) use a combination of feature generation methods as well as machine learning models to predict a business's star rating only from its customers' text reviews. Their approach is to create a bag of words from the top frequent words in all raw text reviews, or top frequent words/adjectives from results of part-of-speech analysis, targeting to remove the bias of stars given by different users. Turney (2002) presents a simple unsupervised learning algorithm for classifying reviews as recommended (thumbs up) or not recommended (thumbs down). The classification of a review is predicted by the average semantic orientation of the phrases in the review that contain adjectives or adverbs, where a phrase has a positive semantic orientation when it has good associations and a negative semantic orientation when it has bad associations.

Maks et al. (2014) focus on the creation of general-purpose polarity lexicons in five languages: French, Italian, Dutch, English and Spanish, using (a) WordNet propagation, a commonly used method to generate these lexicons which gives high coverage of general purpose language and (b) the semantically rich synsets, where concepts are organized in synonym, antonym and hyperonym/hyponym structures which is suited to the identification of positive and negative words. Furthermore, they implement a propagation algorithm and design a method to obtain seed lists, which are similar with respect to quality and size, for each of the five languages. Cieslik (2017a) uses word embeddings and a 2-layers-deep GRU (LSTM) recurrent neural network to compute ratings from texts. Poirier et al. (2010) compute overall ratings from reviews and exploit them to populate a user-item rating matrix to perform CF. Review computation is performed by classification of opinions into two sentiment classes (positive and negative), through machine learning; the trained classifier is also applied to infer ratings from new reviews. Several other implementations for producing ratings from text also exist,[4] but information regarding the techniques they use is not currently available.

The second direction for textual review exploitation identified in Chen et al. (2015) targets the case that both explicit ratings and reviews are available, and in this context reviews act as an auxiliary resource to enhance or refine ratings, providing additional information that can be exploited in the rating prediction or recommendation formulation process. In this context, Raghavan et al. (2012) use the number of upvotes or downvotes of a review to estimate rating helpfulness. Li et al. (2010) use reviews to extract contextual information and use this information to enhance the latent factor model with elements such as time, occasion, location and companion. Cocarascu and Toni (2018) focus on analyzing whether news headlines support tweets and whether reviews are deceptive by analyzing the interaction or the

[4] For instance https://github.com/davelester/Yelp-Rating-and-Review-Trends, https://github.com/beegeesquare/Yelp-star-rating, https://kaggle2.blob.core.windows.net/forum-message-attachments/9420/RecSys%20BrickMovers%20Source%20Code.pdf.

influence that these texts have on the others, thus exploiting contextual information.

In the same line, Aliannejadi et al. (2016) use descriptive keywords extracted from Foursquare tips to compute frequency-based scores, in order to model users' interest and venues and use user's opinions to gain insight on why a particular user liked or disliked a specific place; if some user has not entered a review for a place, the relevant information is derived from other similar users, following the CF principles. The derived reasons for users' likings and dislikings are then exploited in the recommendation formulation process. In their follow-up work, Aliannejadi et al. (2017) further enhance the recommendation process by considering contextual aspects of the visit that has been rated or reviewed; in particular, the season, the travel type (business vs. leisure) and the group type (family, friends, etc.) are taken into account.

McAuley and Leskovec (2013) process review text to detect latent topics within each user's interests and latent topics that pertain to each item and then use the Hidden Factors as Topics (HFT) model introduced in their work for combining the latent factors learned from item ratings with the latent topics learned from reviews. Zhao et al. (2016) also use textual review data from SNs to determine users' latent features. Seroussi et al. (2011) and Musat et al. (2013) exploit review contexts to personalize the ranking of the items in the context of each particular user profile. Maurya et al. (2018) present an algorithm which finds and classifies tweets positive or negative with accuracy toward a specific subject. This proposed system is using the training dataset dictionary to observe the semantic orientation of tweets, in order to know how people feel about an object at a particular moment in time and also tracks how this opinion changes over time. Pero and Horváth (2013) apply a sentiment aggregation method to obtain overall opinions of users toward items, and these opinions are considered together with ratings to address issue of variance in users' bias, i.e., the case that a single user may underrate some items while overrate some other items, compared to her opinion expressed in her reviews.

Contratres et al. (2018) propose a recommendation process that applies sentiment analysis to textual data extracted from Facebook and Twitter and present results of an experiment in which this algorithm is used to reduce the cold start issue. Moshfeghi et al. (2010) utilize emotional features extracted from the movie plot summary and textual reviews, as well as three semantic spaces, namely actor, director, and movie genre to handle the data sparsity problem; the use of emotional features is also found to improve recommendation quality in comparison to the scenario where only the movie space is used. Yang and Fang (2015) use opinions and reviews from online sources to build user profiles and suggestion profiles; these profiles are then used for computing similarities between users and suggestions, and subsequently sort suggestions for a user based on these scores. Finally, Wang et al. (2015) use text descriptions, photos, user check-in patterns from different SNs to enhance the process of computing semantic similarity between venues; this semantic similarity is used for deriving user preferences based on user-venue check-in information.

None of the above-mentioned works considers the issue of the uncertainty introduced by the process of computing ratings based on textual information, and combining the uncertainty-tagged (or, from another viewpoint confidence-tagged) computed ratings with explicitly entered ratings in the rating prediction process. Our work thus advances the state-of-the-art by exploiting textual information from SNs in the context of RSs, both in the stage of rating prediction and in the stage of recommendation formulation. It is worth noting that the work presented in this paper alleviates the cold start problem, by using reviews to infer the ratings that CF systems require, while it can also be combined with other techniques that exploit reviews for tackling the cold start and data sparsity problems, such as (Contratres et al. 2018; McAuley and Leskovec 2013; Musat et al. 2013, Moshfeghi et al. 2010).

3 Rating computation based on textual data and rating prediction

In this section, we present our approach for predicting ratings based on SN textual data. In this context, we gather user-contributed textual data from SNs, and then we use existing approaches to compute a numeric rating from these textual data. For each such computed rating, we calculate a confidence metric, based on the number of positive and negative terms found in the text, following a polarity-based approach (Maks et al. 2014). Subsequently, rating predictions are formulated taking into account (a) explicitly entered metrics, where available, (b) the ratings computed from the SN textual data together with their confidence metrics and (c) the influence factors between users in the SN. In the following paragraphs, we analyze the various phases of the rating prediction process.

3.1 Collecting venue-related user-contributed data from social networks

Within social media, users contribute content mainly by entering textual content and uploading multimedia files. Some of the textual content may contain references to places and signify user purchase and presence, respectively; in this paper, we will use the term *posts* to refer to this information.

The exact way of submitting posts is social media dependent: Facebook currently supports posts, either on the user wall or to other users, where the sending user has the ability to attach a reference to a venue. Additionally, users may post reviews on a venue, where a review contains a numeric rating and, provisionally, some text. All these nodes, in both SNs, are subject to commenting by other users, which can be in turn followed by user responses to the comments and/or other users' further comments, weaving up discussion threads.

Furthermore, a check-in is the basic unit of information contributed by users in the FourSquare Swarm, with the location (venue) being mandatory in the contribution and the text being optional. Similarly, tips in FourSquare are textual passages that are submitted for a particular venue, and are therefore attached to it. Finally, messages on FourSquare always carry along the location of the sender, although this is expressed in absolute coordinates, and reverse geocoding is needed to map these coordinates to a venue. Reverse geocoding can be assisted by the message text for more accurate association between the message and the venue, especially when multiple venues exist in proximity to the user coordinates (e.g., when the user is in a mall, an outlet or in the street, close to multiple venues): In such a context, techniques such as named entity recognition in the message text (Ritter et al. 2011) can prove useful.

The data listed above can be extracted using the relevant SN API calls. In our approach, explicitly entered numeric ratings are used "as is," and a confidence metric of 1.0 is associated with these ratings, indicating certainty about the rating value. On the other hand, textual elements that are contributed by a single user and are related to a venue (either being directly attached to a node associated to a venue or being part of a discussion thread starting from a node related to the item or venue) are concatenated to form a single document; this document is then mapped to a (numeric rating, confidence) pair, as described in the following subsection.

3.2 Calculating numeric ratings and confidence from textual data

Following the collection of the user-contributed, venue-related textual data, the next step is to convert each document (recall that a document contains the text of all posts that have been posted on the SN by a specific user and are associated to some specific venue), to a numeric rating and a relevant confidence metric. To perform the conversion, we use a deep learning-based approach, which employs pre-trained GLOVE word embeddings and a 2-layers-deep GRU (LSTM) recurrent neural network. The dataset used for training is subset of the Yelp Dataset Challenge,[5] containing 150,000 reviews and being balanced with respect to (a) star ratings (each star rating from 1 to 5 had similar number of reviews) and (b) review length. The details for the conversion process and the software implementing the conversion are available at (Cieslik 2017a). For self-containment purposes, we outline here the basic steps of the conversion process; for more details, the interested reader is referred to (Cieslik 2017a, b).

1. GloVe embeddings (Pennington et al. 2014) are computed for the dataset and the weight embedding matrix is prepared. Pre-trained embeddings from pre-trained embeddings from the Glove project[6] are also used in this stage.
2. The dataset is split to testing and training subsets, with the training subset being the 80% of the complete dataset and the testing dataset being the remaining 20%.
3. The weight matrix computed in step 1 is used to create an embedding layer in the RNN, which is implemented on top of the Keras deep learning library (Keras 2018). The RNN also contains two GRU layers with a dimensionality of 100, and two dropout layers with a rate of 0.2; dropout layers are introduced to avoid overfitting.
4. The model is trained using 30 epochs and a batch size of 128.
5. Finally, features are extracted from the last RNN layer and transformed to 2D space using the t-Distributed Stochastic Neighbor Embedding (t-SNE) method (van der Maaten, 2014).
6. For computing rating predictions on the basis of reviews, reviews are tokenized and then a prediction is obtained from the Keras model regarding its rating class (i.e., the numeric rating corresponding to the review).

Other approaches for performing this conversion are available, e.g., (Gregory 2013), and their performance will be explored as a part of our future work.

However, at this stage we should consider that the documents that are fed as input to the conversion process may provide varying degrees of evidence regarding the user stance against the venue: For instance, some documents may have an extremely positive or negative polarity (Maks et al. 2014), in which case adequate evidence exists and therefore confidence to the rating is high, whereas for some other documents polarity may be more neutral, in which case the evidence is weak and consequently the confidence to the review is low. In our work, we consider three alternative ways for computing the confidence to the review:

1. The document length (DL). This approach assigns higher confidence levels to ratings produced by lengthier

[5] https://www.yelp.de/dataset/challenge.

[6] http://nlp.stanford.edu/data/glove.6B.zip.

documents, under the rationale that a lengthier document contains more evidence for the conversion process to take into account.

2. The total number of positive and negative terms within the document (TNPNT). This method assigns higher confidence levels to ratings produced by documents containing more positive or negative terms (e.g., "great," "wonderful," "tasty" are positive terms and "bad," "appalling" are negative terms), under the rationale that these terms provide stronger evidence for the conversion process to take into account. Positive and negative terms are drawn from the opinion lexicon (Liu et al. 2005; Liu 2017). Note that in this case, the presence of negation need not be handled, since both positive and negative terms contribute equally to the document score.
3. The absolute difference between the number of positive and negative terms (ANPNT). This approach assigns higher confidence rankings to ratings produced by documents in which either the positive terms prevail over the negative ones or vice versa, under the rationale that such differences provide clearer evidence regarding the stance of the user toward the venue and avoiding "mixed signals". Positive and negative terms are again drawn from the opinion lexicon (Liu et al. 2005; Liu 2017). In this context, the handling of negation is important, since positive and negative terms contribute differently to the document score. To handle negation, we adopted the approach suggested by Pang et al. (2002) and refined by Chikersal et al. (2015), according to which all tokens between certain negation words and the next punctuation mark are considered to be negated, as long as they are either nouns, adjectives, adverbs or verbs. The list of negation words is drawn from Chikersal et al. (2015), and is as follows: *never, no, noth-ing, nowhere, noone, none, not, havent, haven't, hasnt, hasn't, hadnt, hadn't, cant, can't, couldnt, couldn't, shouldnt, shouldn't, wont, won't, wouldnt, wouldn't, dont, don't, doesnt, doesn't, didnt, didn't, isnt, isn't, arent, aren't, aint, ain't.*

All three approaches have been experimentally tested, by taking into account the probability that some rating computed on the basis of a document having the relevant characteristic (document length; total number of positive and negative terms within the document; absolute difference between the number of positive and negative terms within the document) actually matches ground truth data, which corresponds to the rating explicitly entered by the user in a review (for this experiment, comments from users that had entered reviews were used). The experiment is described in Sect. 5, and it concluded that the third method yields the best results.

3.3 Influence in social networks

Within a SN, "social friends" greatly vary regarding the nature of the relationship holding among them: they may be friends or strangers, with little or nothing in between (Gilbert and Karahalios 2009). Users have friends they consider very close, and know each other in real life and acquaintances they barely know, such as singers, actors and athletes (Shardanand and Maes 1995). Bakshy et al. (2012b) suggest that a SN user responds significantly better to recommendations (e.g., advertisements) that originate from friends of the SN to which the user has a high tie strength. In their work, the strength of the directed tie between users *i* and *j* is linked to the amount of communication that has taken place between the users in the recent past and is computed as:

$$TS_{i,j} = \frac{C_{i,j}}{C_i} \tag{1}$$

where C_i is the total number of communications posted by user *i* in a certain time period (a period of 90 days is considered for computing the tie strength) in the SN, whereas $C_{i,j}$ is the total number of communications posted on the SN by user *i* during the same period and are directed toward user *j* or on posts by user *j*. Although the tie strength metric can be used to locate the influencers of a user, it does not consider user interests, which are important in RS. In our work, we adopt the more elaborate influence metric presented in (Margaris et al. 2018), which computes the tie strength between users *i* and *j* for each distinct interest. In more detail, the influence metric $IL_{i,C}(j)$, where C is an interest category is defined as follows:

$$IL_{i,C}(j) = \begin{cases} TS_{i,j}, & \text{if } C \in interests(i) \wedge C \in interests(j) \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

Effectively, this formula assigns a zero influence level value for interests that are not shared among the considered users, whereas for common interests, the value of the tie strength is used. For the population of each user's interest set, we use the user interest lists collected by the SN (Margaris et al. 2016). Since this list is built automatically when the user interacts with the SN, it will be comprehensive and will include all categories that the user is interested in.

Following the results presented in (Margaris et al. 2018), we consider up to $N=30$ influencers per user and only maintain influencers having an influence level ≥ 0.18. The set of influencers of user u in category C will be denoted as $Infl_C(u)$.

3.4 The taxonomy of venue categories

The influence level calculation scheme presented in Sect. 3.3 relies on the allocation of venues into categories, so as to increase the granularity of the computed influence levels, aiming to increase prediction accuracy and recommendation utility. Venues categorization may be performed at different granularity levels: For example, FourSquare assigns venues to branches of a six-level taxonomy; the following list presents the correspondence between taxonomy levels and relevant information granularities, including also relevant examples:

- level 0: this level encompasses all venues.
- level 1: venue grouping at very high level. Level examples: shop and service, arts and entertainment, nightlife spot, etc.
- level 2: at this level, broad categories of venues are defined. Level examples: shopping mall, museum, bar, etc.
- level 3: at this level, broad-level categories are refined to derive more specific categories. Level examples: accessories Store, science museum, cocktail bar, etc.
- level 4: very detailed classification of venues (available in few level 3 categories only, which are located under the "food" and "outdoors and recreation" level 1 categories). Level examples: Japanese Curry Restaurant (specialization of Japanese restaurants and, Yoga Studio (specialization of Gym/Fitness Center).
- level 5: actual venues.

Margaris et al. (2017) have asserted experimentally that an optimal choice for the categorization detail level is the third level of the above taxonomy (or level 2, where level 3 is unavailable), since (a) categories at this level are adequately specific to provide specialized, category-specific influence levels (b) no overfitting issues occur which would inhibit the computation of category-specific influence levels and (c) the storage space needs for recording user preferences and influence metrics at this level of granularity are limited to less than 100 K per user, which can be accommodated in contemporary systems. Therefore, in this work we adopt employ a level-3 taxonomy to perform venue classification and also store relevant user preferences at this level.

3.5 Rating prediction computation

Having available the user ratings (both explicitly entered and computed), the algorithm proceeds by computing the similarity between users, so as to determine each user's nearest neighbors. Similarity between two users, *u* and *v,* is computed according to formula (3):

$$sim(u,v) = \frac{\sum_{i\in I_u\cap I_v}\left(\left(r_{u,i}-\overline{r_u}\right) * \left(r_{v,i}-\overline{r_v}\right) * c\left(r_{u,i}\right) * c\left(r_{v,i}\right)\right)}{\sqrt{\sum_{i\in I_u\cap I_v}\left(\left(r_{u,i}-\overline{r_u}\right)^2 * c\left(r_{u,i}\right)^2\right)} * \sqrt{\sum_{i\in I_u\cap I_v}\left(\left(r_{v,i}-\overline{r_v}\right)^2 * c\left(r_{v,i}\right)^2\right)}} \tag{3}$$

This is a standard Pearson correlation metric, augmented to take into account the confidence associated with each rating *r*, which is denoted as $c(r)$: according to the modified formula, ratings with higher confidence affect more strongly the computation of the user similarity metric. When all user–user similarities have been computed, we formulate the sets of nearest neighbors $NN(u)$ for each user *u;* each set $NN(u)$ contains the 30 users having the highest similarity with *u*, a setting widely used in user–user CF (Margaris et al. 2018; Liu and Lee 2010; Margaris et al. 2016)

Regarding the computation of the rating prediction, again we modify the standard user–user rating prediction formula, so as to take into account (a) the confidence associated with each rating and (b) the influence levels between users; the modified formula is shown in Eq. (4):

$$p_{u,x} = \overline{r_u} + \frac{\sum_{v\in NN(u)} sim(u,v) * w_{u,Cat(x)}(v) * c\left(r_{v,i}\right) * \left(r_{v,i}-\overline{r_v}\right)}{\sum_{v\in NN(u)}\left|sim(u,v) * c\left(r_{v,i}\right) * w_{u,Cat(x)}(v)\right|} \tag{4}$$

In this formula, again $c(r)$ denotes the confidence assigned to rating *r*, whereas $w_{u,Cat(x)}(v)$ corresponds to the weight associated with the opinion of user *u* in relation to *v* for the category that item *x* (i.e., the item for which the prediction is computed) falls in. Similarly to the approach presented in (Liu and Lee 2010), the weight is used to amplify the effect that a user's influencers have on the computation of the predictions and is defined as follows:

$$w_{u,Cat}(v) = \begin{cases} 1 + IL_{u,Cat}(v), & \text{if } v \in Infl_{Cat}(u) \\ 1, & \text{if } v \notin Infl_{Cat}(u) \end{cases} \tag{5}$$

adopting the formula used in the *hybrid* approach presented in (Liu and Lee 2010), which is the best performing one among the options reviewed in that work, but substituting the item category-insensitive tie strength between users with the category-aware influence level discussed in Sect. 3.3.

4 Recommendation formulation

Most recommender systems compute the recommendations offered to users by initially predicting the rating that a user would assign to each non-rated item, and subsequently selecting the items having the top-K predicted ratings. In the domain of venue recommendation formulation, however, additional parameters have to be considered: one such factor is *proximity*, since venues in close distance are more bound to be useful than distant ones. Margaris et al. (2017) establish that *quality of service parameters* (ITU 1998) such as cost, service quality and atmosphere, play an important role in the utility of venue recommendation: indeed, if a user typically dines in restaurants with an average check per person in the range of \$30-\$40, it would not be suitable to offer a recommendation for a restaurant with an average check per person equal to \$150, due to the fact that some of the users' influencers for the specific restaurant category have rated or commented favorably on the specific restaurant. A more appropriate approach would be to offer a recommendation for a restaurant that is "similar" to the one having the \$150 check per person, but being at the price range of \$30–\$40, to match the preferences of the user for which the recommendation is formulated.

The notion of similarity referenced above spans across multiple aspects. In this paper we will consider two aspects sourced from the bibliography, namely *semantic venue similarity* (Margaris et al. 2017) and *physical distance-based venue similarity* (Jones et al. 2001). We note here that the design of the recommendation algorithm presented below allows for more aspects to be accommodated, to broaden the range of criteria taken into account.

In the following, we firstly describe briefly venue QoS parameters as well as the aspects of semantic venue similarity and physical distance-based venue similarity, in order to promote self-containment. Subsequently, we introduce an algorithm that formulates recommendations within a SN, by synthesizing (a) the rating predictions computed by the algorithm presented in Sect. 3 and (b) quantifications of QoS parameter-based similarity, semantic similarity and physical distance-based and thematic-based venue similarity.

4.1 QoS parameters for venues

QoS is typically defined through attributes (ITU, 1988). While a multitude of attributes that can be used for expressing a venue's QoS exist (Mersha and Adlakha 1992), in this paper will consider only the attributes cost (c), service (s) and atmosphere (a). This set of attributes is employed by many major travel services and websites, including TripAdvisor (http://www.tripadvisor.com) OpenTable (https://www.opentable.com/); furthermore, the extension of the algorithm to include additional attributes is straightforward; hence, confining the discussion to these three attributes does not lead to loss of generality.

In order to decide on which venue to visit, a user is expected to aim toward the maximization of service and atmosphere and the minimization of cost; since these goals are typically contradictory, a "golden cut" would be pursued by, e.g., compromising cost optimality in favor of obtaining higher service level. Cost is usually expressed in actual currency, while a normalized indicator may also appear (e.g., from a single dollar sign for very cheap venues to five consecutive dollar signs, for very expensive ones); service and atmosphere are expressed in some scale, typically 1–5 or 1–10. In the rest of this paper we will use actual currency to represent cost and adopt the scale 1–10 for service and atmosphere. An example of values concerning the London's restaurants qualitative characteristics are shown in Table 1 (values are sourced from TripAdvisor[7]).

Table 1 Sample QoS values within the repository

Place	Cost	Service	Atmosphere
Restaurant Gordon Ramsay	\$140	10	9
Italian Pizza Connection	\$45	9	8
London Fish and Chips	\$8	8	7
…			

4.2 Venue semantic information and similarity

The semantic information of venues can be accommodated using ontologies (Margaris et al. 2017). Under this approach, the taxonomy described in Sect. 3.4 is enriched as follows:

- Nodes representing categories, which form a tree by virtue of the fact that these nodes form a taxonomy, are enhanced with a set of property definitions, which are applicable to all venues that are classified in the particular category (or any more specific one). A property definition lists the property name and type (e.g., integer, string, enumeration etc.). For example, the category "Nightlife spots" may specify a property "Capacity" of type "integer," which would be applicable to all actual venues belonging in this category or any of its subcategories.
- Each leaf node may use any of the properties applicable for its category, and populate it with a specific value, compatible with the type of the property.

Having this representation available, the semantic similarity $semSim(v_i, v_j)$ between two venues v_i and v_j can be computed as follows (Margaris et al. 2017):

$$semSim(v_i, v_j) = \frac{\sum_{p \in v_i \wedge p \in v_j} sim_p\left(p(\mathrm{v_i}), \mathrm{p}(\mathrm{v_j})\right)}{\left|prop(v_i) \cup prop(v_j)\right|} \tag{6}$$

[7] http://www.tripadvisor.com/Restaurants-g186338-London_England.html.

where $p(v_i)$ and $p(v_j)$ are the values of property p for venues v_i and v_j, respectively; $sim(p(v_i), p(v_j))$ is a metric of the similarity between the values of property p; and finally, $prop\left(v_i\right)$ (resp. $prop\left(v_j\right)$) is the set of properties in venue $\mathrm{v_i}$(resp. $\mathrm{v_j}$). Note that the similarity computation function is property-specific; for instance, when comparing the attribute *musicGenre* for two venues, $sim_{musicGenre}(newWave,\ postPunk)$ mayyield 0.9 (i.e., a high value) and $sim_{musicGenre}(newWave,\ opera)$ may yield 0.1 (i.e., a low value). For numeric-typed properties such as ratings and costs, the sim_p function may be defined as:

$$sim_{num\_prop}\left(v_1, v_2\right) = 1 - \frac{\left|v_1 - v_2\right|}{\max(num\_prop) - \min(num\_prop)} \tag{7}$$

where $max(num\_prop)$ and $min(num\_prop)$ are the maximum and minimum values, respectively, of *num_prop* in the ontology extension. Equation (7) effectively corresponds a numeric value normalization formula (Aslam and Montague 2001). Domain-specific similarity functions can be employed to leverage similarity calculation accuracy, e.g., Pirasteh et al. (2014) introduce methods for computing metrics sim_g and sim_d, representing the similarity between movie genres and movie directors, respectively. If Eq. (7) cannot be used and no domain-specific similarity is available, Eq. (8) can be employed as a fallback similarity computation formula.

$$sim_{default}(v1, v2) = \begin{cases} 1, & \text{if } v1 = v2 \\ 0, & \text{otherwise} \end{cases} \tag{8}$$

For a more detailed discussion on venue similarity computation, the interested reader is referred to (Margaris et al. 2017).

4.3 Physical distance-based venue similarity

Jones et al. (2001) have demonstrated that the physical distance between venues plays an important role, since venues in close proximity are more bound to be visited by the same person, contrary to venues that are distant from each other. The computation of the physical distance-based venue similarity between the venues in (Jones et al. 2001) takes into account two factors, namely the normalized Euclidian (NED) and the normalized hierarchical distance (NHD): NHD is based on the "part-of" relation hierarchy(e.g., Wisconsin is *part-of* Dane County, which is *part-of* the State of Wisconsin, which is *part-of* the U.S.A., etc.). The two metrics are combined into a single, comprehensive metric denoted as *Total Spatial Distance* (*TSD*) using a weighted sum approach, as denoted in Eq. (9):

$$TSD\left(loc_1, loc_2\right) = w_e * NED\left(loc_1, loc_2\right) + w_h * NHD\left(loc_1, loc_2\right) \tag{9}$$

In Eq. (9), w_e and w_h represent the weights assigned to *NED* and *NHD*, respectively; Jones et al. (2001), w_e is set to 0.6 and w_h to 0.4. Based on the TSD metric (which is normalized in the range [0, 1]), we can compute physical distance-based venue similarity as

$$PhysDistSim\left(loc_1, loc_2\right) = 1 - TSD\left(loc_1, loc_2\right) \tag{10}$$

For more details, the interested reader is referred to Jones et al. (2001); note that in this work, a *venue thematic distance metric* is also used, however in our work thematic distance is encompassed into the semantic similarity metric.

4.4 Venue recommendation formulation

In order to formulate, a venue recommendation that considers on the one hand the opinions of the users' nearest neighbors and influencers, and on the other hand QoS and similarity aspects, two subtasks are executed in parallel, following the RS architecture presented in (Margaris et al. 2017). In particular:

1. the first task computes a QoS-based recommendation considering only the qualitative characteristics of each venue, and
2. the second task computes a CF-based recommendation considering the opinions of the user's nearest neighbors and influencers. In this context, for each place category, we use a distinct set of influencers (which are stored in the user profile), aiming to leverage the accuracy of recommendation (Margaris et al. 2016, 2017).

Afterward, the two recommendations are combined to formulate the final recommendation, employing a metasearch algorithm (Aslam and Montague 2001), as presented in Margaris et al. (2015).

For each venue, the QoS- and CF-based recommendation scores regarding user u ($score^{QoS}_{v,u}$ and $score^{CF}_{v,u}$, respectively) are combined into a single recommendation score, through the application of the $WCombSUM_i$ formula (He and Wu 2008). The $WCombSUM_i$ formula effectively computes the overall score $VenueScore_{v,u}$ as the weighted sum of the two partial scores $score^{QoS}_{v,u}$ and $score^{CF}_{v,u}$. More specifically, the overall score for a venue v within the final recommendation for user u is calculated as:

$$VenueScore_{v,u} = \mathrm{w}^{CF}_{C(v),u} * score^{CF}_{v,u} + \mathrm{w}^{QoS}_{C(v),u} * score^{QoS}_{v,u} \tag{11}$$

where $C(v)$ denotes the category of venue v; $\mathrm{w}^{CF}_{C(v),u}$ and $\mathrm{w}^{QoS}_{C(v),u}$ are weights assigned to the scores produced for venue v by the CF-based and the QoS-based algorithm, respectively.

To further promote tailoring of recommendation to individual users, the weights $w^{CF}_{C(v),u}$ and $w^{QoS}_{C(v),u}$ are both user-specific and category-specific, e.g., the weight used for the category *museums* may be different for users u_1 and u_2, while additionally the weights used for recommending a bar to user u_1 may be different than the ones employed when recommending a shopping mall to the same user.

Weight value computation is based on the assessment of how receptive a user is to the recommendations made by her influencers: the greater the receptiveness level, the higher the weight assigned to the CF-based dimension. More specifically, the values of the weights are calculated as follows:

$$w^{CF}_{c,u} = \frac{\left|VenuesVisitedDueToInfluence_{c,u}\right|}{\left|VenuesVisited_{c,u}\right|}$$

$$w^{QoS}_{c,u} = 1 - w^{CF}_{c,u} \tag{12}$$

We can observe that the CF-based score weight $w^{CF}_{c,u}$ is calculated as the ratio of the number of venues within category c that user u has visited due to recommendations made to her based on her influencers' or near neighbors ratings, by the total number of places within category c that u has visited. Obviously, a ratio value close to 1 indicates that the user nearly always follows these recommendations, while a value close to 0 denotes that influencers' recommendations are disregarded by the user. In order to estimate the set $VenuesVisitedDueToInfluence_{c,u}$, we adopt the approach introduced by Margaris et al. (2017), according to which a visit to a venue v by a user u is deemed to have been triggered by the user's influencers or near neighbors if (a) the system had offered to the user a recommendation for the venue prior to her visit (b) the recommendation had considered the rating entered by an influencer or near neighbor.

The computation of the CF-based score ($score^{CF}_{v,u}$) and the QoS-based ($score^{QoS}_{v,u}$) referenced in Eq. (10) is described in the following paragraphs. The operation of the algorithm is divided in three phases: (a) *offline initialization*, where a set of metrics required for recommendation formulation is pre-computed and stored in a database to promote efficiency (b) *online operation*, where recommendations to users are formulated and (c) *repository update*, where changes in the SN status and the venue database are accommodated into the pre-computed metrics database, by recomputation of the affected metric values.

Phase 1—offline initialization The bootstrapping of the algorithm entails the following actions:

- for each venue category c, the minimum and maximum values for all the QoS attributes among all venues in the category are computed. The equations used for the computation of the minimum and maximum cost within a category c are shown in Eq. (13), while the calculation of the minimum and maximum service and atmosphere within a category C is performed in an identical fashion.

$$\begin{aligned} minCost(c) &= \min_{venue_i \in c}\left(cost\left(venue_i\right)\right) \\ maxCost(c) &= \max_{venue_i \in c}\left(cost\left(venue_i\right)\right) \end{aligned} \tag{13}$$

- for each user u and venue category c:
  - the CF-based and QoS-based weight values ($w^{CF}_{c,u}$ and $w^{QoS}_{c,u}$, respectively) are computed, by employing Eq. (12).
  - the average QoS values (cost, service and atmosphere) of the venues within category c that user u has visited in the past are computed.
  - the level of influence of her social friends for the particular category is calculated as discussed in subsection3.3, and subsequently the *top-K* ones with the highest influence levels are retained. Regarding the value of K, in this paper we use the value $K=6$, adopting the results of (Margaris et al. 2016) which demonstrate that this setting yields optimal results.
- for each user u, the venues that she has visited are stored in her profile using a taxonomy level-3 detail.
- for each pair of venues (v_1, v_2), we compute their similarity $VenueSim\left(v_1, v_2\right)$,considering (a) the semantic similarity between (v_1, v_2) and (b) the physical distance-based similarity between the venue locations; the similarity between venues v_1 and v_2 is computed as:

$$\begin{aligned} &VenueSim\left(v_1, v_2\right) \\ &\quad = SemSim\left(v_1, v_2\right) * PhysDistSim\left(loc\left(v_1\right), loc\left(v_2\right)\right) \end{aligned} \tag{14}$$

where $SemSim(v_i, v_j)$ is the semantic similarity between venues v_i and v_j (c.f. Sect. 4.2, $loc(p)$ denotes the location at which p is located, and $PhysDistSim(loc_i, loc_j)$ corresponds to the physical distance-based similarity of locations loc_i and loc_j.

Phase 2—online operation Once initialization has concluded, the online operation phase of the algorithm commences, during which recommendations are generated. Algorithm execution is triggered when a recommendation for a user U regarding venues in a category C is needed: this may be due to an express request from the user for such a recommendation, or when the SN logic considers the forwarding of such a recommendation to be appropriate.

Recommendation formulation proceeds by first computing rating predictions for all venues in category C that U has not visited insofar. For each of these venues w, the respective QoS-based scores $score^{QoS}_{w,U}$ are computed:

$$score^{QoS}_{w,U} = cost\_vicin(U, w) * ser\_vicin(U, w) * atm\_vicin(u, W) \tag{15}$$

where $cost\_vicin(U,\ v)$ (cost vicinity) quantifies how close the venue price is to the user's price habits within the specific category. This is computed as

$$cost\_vicin(U, w) = 1 - \frac{|cost(w) - MC(U, C)|}{maxCost(C) - minCost(C)} \tag{16}$$

where $cost(w)$ is the cost associated with venue w and $MC(u,C)$ corresponds to the mean cost of places within category C that U visits. Correspondingly, the calculation of service vicinity and atmosphere vicinity is illustrated in Eq. (17):

$$ser\_vicin(U, w) = \begin{cases} 1 - \frac{|ser(w) - MS(U,C)|}{maxSer(C) - minSer(C)}, & \text{if } ser(w) \leq MS(U, C) \\ 1, & \text{if } ser(w) > MS(U, C) \end{cases}$$
$$atm\_vicin(U, w) = \begin{cases} 1 - \frac{|atm(w) - MA(u,C)|}{maxAtm(C) - minAtm(C)}, & \text{if } atm(w) \leq MA(U, C) \\ 1, & \text{if } atm(w) > MA(U, C) \end{cases} \tag{17}$$

where $MS(U,C)$ *and* $MA(U,C)$ are the mean service and mean atmosphere, respectively, of places visited by U within C, and $ser(w)$ and $atm(w)$ are the service and atmosphere ratings. In formula (17) we can observe that when the actual value of a venue's service or atmosphere surpasses the mean value of the respective metric for the particular user and venue category, the venue is considered as totally similar to the user's profile: this stems from the fact that users always try to maximize service and atmosphere.

If the value of $score^{QoS}_{w,U}$ surpasses a pre-specified threshold Th_{QoS}, then the QoS parameters of venue w are deemed to be adequately close to the QoS levels of venues typically visited by U; in this respect, w is marked as a candidate for recommendation. In this respect, its overall score is computed by employing formula (11), and venue v along with its overall score is stored in the "potential recommendations" list. In this work, we use the threshold value $Th_{QoS} = 0.68$, adopting the results of (Margaris et al. 2017).

If, the QoS-based score $score^{QoS}_{w,U}$ is less than the Th_{QoS} threshold value (0.68), then the QoS parameters of w are deemed to be "not close enough" to venue visiting patterns of user U within category c, and therefore w is considered as not appropriate for recommendation. Taking this into account, the algorithm proceeds to find a venue z which (a) satisfies the QoS requirements of user U and (b) is "similar" to w. More specifically, following steps are taken:

1. the algorithm locates within category c venues z for which $score^{QoS}_{z,U}$ is greater than the threshold value $Th_{QoS} = 0.68$. Since the QoS-based score for these venues is greater than the threshold, they can be candidates for recommendation to U.
2. For each such venue z, the respective CF-based rating is computed as shown in Eq. (18):

$$score^{CF}_{z,U} = score^{CF}_{w,U} * PhysDistSim(loc(w), loc(z)) \\ * SemSim(w, z) \tag{18}$$

In Eq. (18), we can observe that the CF-based score value for the "replacement" venue z starts off with the CF-based score value of the original venue w and is subsequently attenuated through the consideration of physical distance and semantic (dis) similarities between w and z.

3. Finally, all venues z identified in step 2 are considered: the one having the highest score is retained and appended to the list of potential recommendations.

When all candidate venues have been examined, the K items having the top-K overall scores are extracted from the list of potential recommendations and are recommended to the user; number N may vary, depending on the system settings.

Phase 3—repository update The dynamic nature of the content of the SNs and venues information, a number of database elements and linkages need to be updated, so as to keep the database up-to-date. The cases when a database update is needed are defined below:

The updates that need to be performed are as follows:

1. Each time a new venue is stored in the venue database, the minimum and maximum QoS values for all QoS attributes within this category may need to be updated.
2. When a user check-in of a user U to a venue within category C is posted to the SN, the mean QoS attribute values of the set of places within category C that user U has visited need to be updated.
3. When a user checks in a new place, this modifies the set of places that the user has checked in; if I the check-in was triggered by a recommendation to which an influencer has contributed, then the set of places visited due to influence is also modified.
4. Each time a new venue x is stored in the venue database, the similarity between x and all other venues within the database need also to be computed.

5. Finally, when a user's categories of interest change (typically when a user visits a place belonging to a category that she has not checked-in before) or the number of communications between the user and her social acquaintances is modified, the *top-K* influencers of each user *u* within each category of interest *C* need to be computed anew.

Updates (1) and (2) are computationally inexpensive, therefore they can be performed synchronously line with the processing of the triggering event. On the other hand, steps (3)–(5) are more computationally demanding; to this end, they can be executed in batch fashion, e.g., be executed periodically.

5 Experimental evaluation

In this section, we report on our experiments aiming to:

(a) explore the correlation between textual review/comment features and rating prediction accuracy, in order to identify the most prominent feature to be used for the computation of review confidence
(b) determine the optimal weights assigned to the implicit ratings, that are produced by the users' textual reviews and comments on the items to be recommended and
(c) evaluate the performance of the proposed approach, in terms of prediction accuracy and users' satisfaction regarding the offered venue recommendations.

5.1 Exploring review features

In this section, we explore which of the three textual review features presented in Sect. 3.2, can be associated with the improvement of rating prediction accuracy. To validate the existence of such an association, we conducted an experiment where for each feature the following procedure was used:

(1) for each test dataset *D*, containing (*user Id, itemId, rating, textualReview*) tuples, initially, we converted all textual reviews in each dataset to ratings, and then calculated the MAE of this conversion process, comparing the computed ratings with the explicitly entered ones, i.e.,

$$MAE = \frac{1}{n} \sum_{r_{u,i} \in RatingsDB} \left| r_{u,i} - \hat{r}_{u,i} \right| \tag{19}$$

where $r_{u,i}$ is an explicitly entered rating (a "ground truth" value) in the ratings database, and $\hat{r}_{u,i}$ is the value computed for that rating on the basis of the textual review.

(2) subsequently, for each dataset we iteratively increased a threshold value for the particular feature, dropping at each iteration those reviews in the dataset that had a feature value lower than the threshold value; for example, when the document length feature was examined and the document length threshold value was set to 10, we dropped from the dataset all reviews consisting of less than 10 words. Then, we converted the remaining textual reviews in the dataset to ratings, and we calculated the MAE in the same fashion as in the previous step.

The rationale behind this approach is that if higher values for a feature (e.g., document length) are associated with increments in prediction accuracy, then dropping from the dataset reviews with low values for the particular feature (e.g., reviews with a small length) should lead to the computation of more accurate ratings and henceforth, the MAE would drop.

Regarding the three features listed in Sect. 3, the following thresholds were considered in the experiment iterations:

- For the *document length* (DL) characteristic, we initially considered all documents, while in the second iteration only documents for which DL ≥ 10 were considered. In the subsequent iterations, DL thresholds of 20, 30, 40 and 50 were used.
- For the *total number of positive and negative terms within the document* (TNPNT) characteristic, the first iteration again contained all documents, while in the second iteration only documents for which TNPNT ≥ 2 were taken into account. Afterward, we proceeded using the TNPNT threshold values of 5, 10, 15 and 20.
- For the *absolute difference between the number of positive and negative terms within the document* (ANPNT) characteristic, the first iteration involved all documents, while in the second iteration we considered only documents for which ANPNT ≥1. In subsequent iterations, the ANPNT threshold values of 2, 3, 4, 5, 6 and 7 were used.

For this experiment we used a machine equipped with six Intel Xeon E7 4830 @ 2.13 GHz CPUs, 256 GB of RAM and one 900 GB HDD with a transfer rate of 200MBps, which hosted the datasets and ran the rating prediction algorithms.

In order to produce reliable results, we applied our experiment on six Amazon datasets (musical instruments; automotive; clothing, shoes and jewelry; sports and outdoors; electronics and cell phones and accessories). We chose these datasets since they are comprehensive and reliable datasets,

Table 2 Datasets summary

Dataset name	#users	#items	#reviews	DB size (in text format)
Automotive	2.9 K	1.8 K	20.4 K	14.0 M
Cell phones and accessories	27.8 K	10.4 K	194.4 K	136.0 M
Clothing, shoes and jewelry	39.4 K	23.0 K	278.6 K	147.0 M
Electronics	192.4 K	63.0 K	1.6 M	1.5 G
Musical instruments	1.4 K	0.9 K	10.2 K	7.2 M
Sports and outdoors	35.5 K	18.3 K	296.3 K	199.0 M

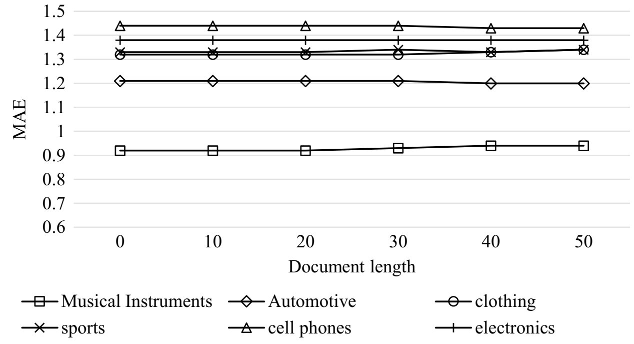

Fig. 1 MAE against document length for the Amazon datasets

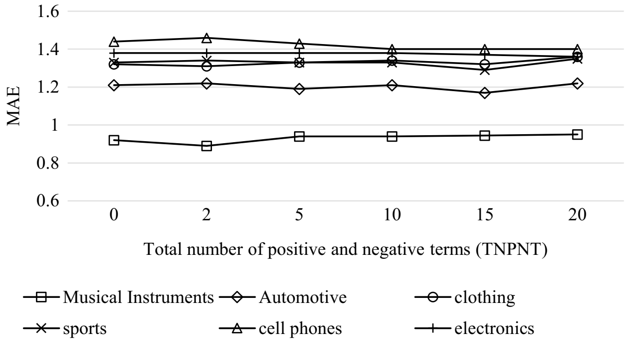

Fig. 2 MAE against total number of positive and negative terms (TNPNT) for the Amazon datasets

containing user reviews and ratings from various domains; furthermore, they are widely used in RS research. The properties of these datasets are listed in Table 2; please note that all datasets are 5-core, i.e., they contain only users that have entered at least 5 reviews, and items that have been reviewed by at least 5 users. We chose to use only datasets in which items and their descriptions are sentiment-neutral, and hence positive or negative terms within the reviews are more bound to be related to user satisfaction. In other cases, positive and negative terms may not refer to user satisfaction but to the product features (e.g., the term "scary" is generally negative; however, in the review of a horror novel or a thriller movie, it can be considered to be positive). In our future work, we plan to utilize methods such as those presented in (Moshfeghi et al. 2010), which take into account the emotional features extracted from the movie plot summary and textual reviews, so as to more accurately determine the actual user rating.

For each dataset, we calculated predictions by randomly selecting each time a rating, hiding its numeric value, trying to predict the numeric value from the textual rating and computing the difference between the hidden rating and the predicted value. This procedure was performed multiple times, considering at least five ratings for each user and testing at least 25% of the total number of reviews; finally, the MAE was calculated based on the differences between the hidden and predicted ratings.

Figure 1 illustrates the results obtained from the experiments on the Amazon datasets, regarding the relation between the MAE and the document length feature. We can observe that the MAE is practically not affected by the variations of the document length: only small changes are noted ranging from −0.83 to 2.17% (note that a negative change indicates that the MAE drops and thus improves, while an increment in the MAE signifies a deterioration). It is worth noting that the biggest changes are observed in the musical instruments dataset (2.17% for document lengths 40 and 50) and the automotive dataset (−0.83% for document lengths 40 and 50), with these two datasets being those with the smallest number of reviews among the six tested datasets. On the contrary, the electronics dataset, which has the largest number of reviews, exhibited a constant MAE for all tested document lengths.

Figure 2 depicts the results obtained from the experiments on the Amazon datasets, regarding the relation between the MAE and the total number of positive and negative terms feature. Here the changes in the MAE range between −3.31% and 3.26%, with the largest drop (improvement) being identified in the case of the automotive dataset (the smallest one) when TNPNT = 15; notably, when TNPNT increases to 20 in the same dataset, the MAE obtained is marginally worse that the MAE obtained for TNPNT = 0 (an increment of 0.8%), indicating the instability of this dataset, due to the small number of reviews.

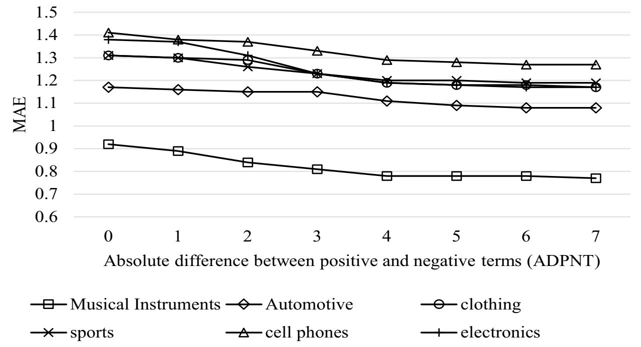

Fig. 3 MAE against absolute difference between positive and negative terms (ADPNT) for the Amazon datasets

Figure 3 presents the results obtained from the experiments on the Amazon datasets, regarding the relation between the MAE and the absolute difference between positive and negative terms (ADPNT) feature. Similarly to the behavior observed in Fig. 3, the MAE metric drops when ADPNT increases up to the value of 4; in two of the datasets the MAE drop is comparable to the one observed in Fig. 3 (electronics and musical instruments, with the respective MAE drops being 13.8% and 15.2%, respectively), while in the remaining datasets the drop ranges from 7.7% (automotive) to 9.2% (clothing). Further increments in the value of ADPNT have small or no effect on the MAE (changes range from 0% in the case of the automotive dataset to 1.2% in the case of the cell phone dataset).

Conclusively, our experiments with the Amazon datasets validate that (a) the ADPNT feature is positively correlated with the confidence of the computed ratings, and (b) computed ratings that are based on documents having ADPNT ≥ 4 can be considered as reliable as explicitly entered ratings, while reviews computed on the basis of documents having ADPNT < 4 will be assigned a confidence level less than 1.

5.2 Determining the implicit information's ratings' confidence

The second experiment is aimed at determining the exact confidence level to be assigned to ratings produced by documents. In this experiment, we have examined all combinations of confidence level assignments to values of ADPNT, in order to identify the combination that provides the best prediction accuracy, i.e., it minimizes the MAE. More specifically, for each of the ADPNT values (0, 1, 2, 3, 4) the confidence value assignments of 0.1, 0.2, …, 0.9, 1 were considered, and all possible assignment combinations were used with the rating prediction algorithm presented in Sect. 3; for each combination, after rating predictions were formulated, the MAE was computed. In all cases, explicitly entered numeric ratings and ratings produced on the basis of documents with ADPNT > 4 were assigned a confidence level of 1.

Figure 4 displays the 10 confidence level assignment settings which produced the smallest MAE. In each column, setting the value of the "ADPNT = 0" series corresponds to the confidence level assigned to reviews for which the ADPNT metric is equal to zero, and similarly

Fig. 4 Evaluating weight assignments to ADPNT value classes

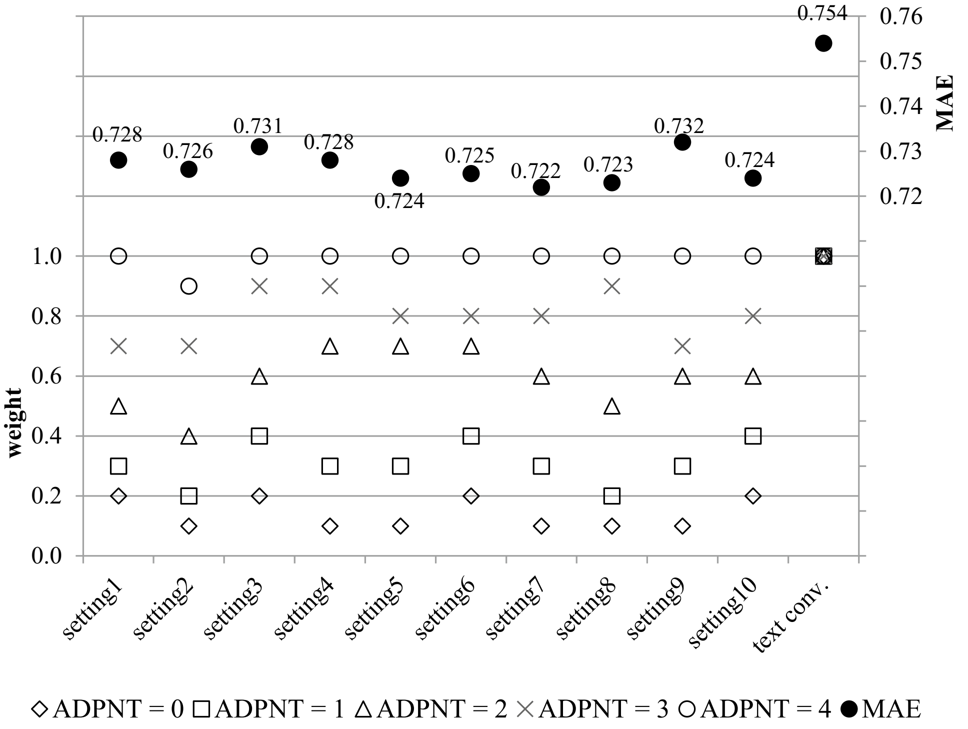

Table 3 Results regarding confidence level assignment for the six Amazon datasets

Dataset name	Setting yielding the minimum MAE	Ranking of setting #7	MAE of setting #7 in relation to minimum MAE (%)
Automotive	Setting #10	3	100.24
Cell phones and accessories	Setting #5	2	100.18
Clothing, shoes and jewelry	Setting #7	1	100.0
Electronics	Setting #7	1	100.0
Musical instruments	Setting #10	2	100.15
Sports and outdoors	Setting #7	1	100

for series "ADPNT = 1," "ADPNT = 2," "ADPNT = 3," "ADPNT = 4." The minimum MAE (0.722; the scale of the MAE is depicted on the right y-axis) is achieved by setting #7, closely followed by setting #8 (MAE = 0.723) and setting #10 (MAE = 0.724). Taking into account that when computing predictions using the ground truth ratings (i.e., using the ratings explicitly given by the experiment participants, and hiding one which was then calculated and compared to the hidden value) the MAE was 0.721, we can conclude that all these settings achieve predictions very close to the optimal. Figure 4 also displays the MAE achieved by employing a plain "text to rating" conversion (last column); in that column all ratings produced by the "text to rating" conversion are assigned a confidence level of 1. We can notice that the MAE associated with that setting is 0.754, which is 4.43% higher than the MAE achieved by setting #7 (i.e., the best performing one).

We validated the results obtained from our study by running the same experiments for the six Amazon datasets listed in Table 2; again, for each dataset, we created a mixture of (a) explicit (numeric) ratings and (b) implicit ratings, produced by user reviews. Then, we examined all combinations of confidence level assignments to values of ADPNT, computing the MAE for each combination.

Table 3 illustrates the results obtained: Setting #7, which produced the smallest MAE in our user study, also produced the smallest MAE in three Amazon datasets (clothing, shoes and jewelry; electronics; sports and outdoors). In two other datasets (cell phones and accessories and musical instruments), setting #7 was the runner up, and in one dataset it was ranked third, however the MAE it produced in these three cases was only marginally worse than the MAE produced by the winning setting (MAE deterioration was found to be less than 0.24% in all cases).

Considering the results of this experiment, the confidence level assignments for ADPNT value classes used in further experiments are drawn from setting #7 and are as follows: $c(ADPNT = 0) = 0.1;$ $c(ADPNT = 1) = 0.3;$ $c(ADPNT = 2) = 0.6;$ $c(ADPNT = 3) = 0.8;$ and $c(ADPNT = 4) = 1$.

5.3 User satisfaction

After having established the optimal coefficient settings for the operation of the proposed algorithm, we run a third experiment, aiming at assessing the participants' satisfaction regarding the recommendations they received from the algorithms presented in Sects. 3 and 4, and at comparing this satisfaction level to that obtained from other related algorithms.

To assess recommendation quality, we conducted a user survey in which 60 people participated. The participants were students and staff from the University of Athens community, coming from 4 different academic departments (computer science, medicine, physics and theater studies). 29 of the participants were women and 31 were men, and their ages range between 18 and 54 years old, with a mean of 29. All of the participants have been Facebook users for at least three years, using it for at least 4 days a week and 1 h of use per day, and we extracted the profile data needed for the algorithm operation using the Facebook Graph API.[8] Regarding the participants' profile and behavior within Facebook, the minimum number of Facebook friends among the participants was 71 and the maximum was 629 (with a mean of 228). For each person, we computed the relevant tie strengths with all of her Facebook friends in an offline fashion.

The venues data used in the experiment were extracted from TripAdvisor. The data set consisted of 5000 places in 20 cities (including New York, London, Rome, Paris and Athens) and falling in 10 places categories (museums, religious/historical monuments, bars, nightclubs, cinemas, theaters, fast food restaurants, cafés and restaurants). The cost attribute values in this repository were set according to the places' current prices, while the service and atmosphere attribute values were set according to the users' rating summary from TripAdvisor. The experiment data are available in (Margaris and Vassilakis 2018a); the textual elements of the comments however are not included in the dataset, because users did not consent for their public availability (most of the

[8] https://developers.facebook.com/docs/graph-api.

textual elements are posted with access limited to specific groups).

In order to quantify and highlight the benefits of introducing and exploiting (a) the confidence factor in textual review to rating conversion and (b) the influence between users derived from the social network, we have compared the proposed algorithm against the following ones:

1. A variant of the proposed algorithm, where all ratings calculated from textual reviews were assigned a value of 1 (i.e., the confidence factor was not used),
2. the algorithm presented in (Margaris et al. 2017), which takes into account social influence, QoS aspects, venue semantic similarity and venue physical distance, but relies solely on explicitly entered ratings.

In this experiment, each participant was asked to rate 18 venue recommendations presented to her, on a scale of 1 (totally unsatisfactory) to 10 (totally satisfactory). The 18 recommendations assessed by each user were generated using the three above listed algorithms, with each algorithm having generated 6 of the recommendations. The recommendations offered to the users covered 90% of the taxonomy level-2 category of places (from bars, pizza places and museums, to casinos and zoos). Recommendations were presented to the users for assessment in randomized order. If two algorithms recommended the same venue, then the venue appeared only once in the result set presented to the user. The experiment data are available in (Margaris and Vassilakis 2018a); the textual elements of the comments however are not included in the dataset, because users did not consent for their public availability (most of the textual elements are posted with access limited to specific groups).

For this experiment, we used two machines. The first was equipped with one 6-core Intel Xeon E5-2620@2.0 GHz CPU and 16 GB of RAM, which hosted the processes corresponding to the active users (browser emulators), i.e., the users who generated the triggering events. The second machine's configuration was identical to the first, except for the memory which was 64 GB; this machine hosted (i) the algorithm's executable, (ii) a database containing the users' profiles including the influence metrics per category, the lists of top *N* influencers per category and the data regarding the posts made by each user and (iii) the venues database, which includes their semantic information and QoS data (cost, reliability, service and atmosphere). The machines were connected through an 1 Gbps local area network.

Figure 5 depicts the participants' satisfaction regarding the recommendations they received, as measured in this experiment. On average (last column on Fig. 5) it is clear that the proposed algorithm outperforms the other algorithms, attaining an overall user satisfaction of 8.45.

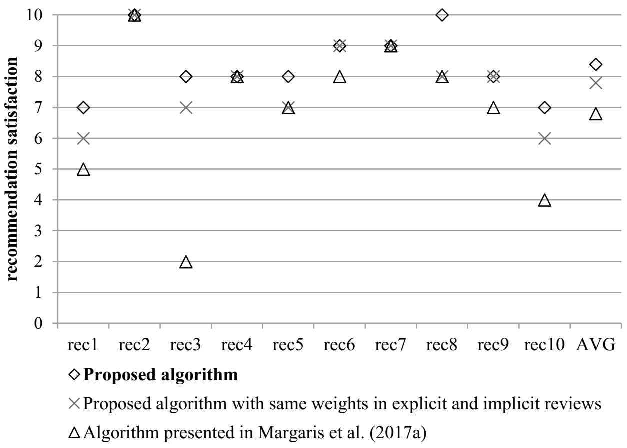

Fig. 5 Users' satisfaction of recommendations made by individual recommendation algorithms

The satisfaction scores assigned by users to the recommendations generated produced by the proposed algorithm range from 4 to 10, with a median of 9 and standard deviation equal to 1.45; 85.4% of the scores fall in the range [7, 10]. The proposed algorithm setting the same weight in explicit and implicit reviews comes in second with an overall user satisfaction of 7.8, falling short of the proposed algorithm's performance by 8.3%. This indicates that the introduction and exploitation of the confidence levels which are attached to ratings that are computed on the basis of textual reviews offers a considerable performance improvement.

The algorithm presented in (Margaris et al. 2017), i.e., the algorithm that relied only on explicitly entered ratings, was ranked in the last place with an overall user satisfaction of 6.8. The main cause of dissatisfaction from this algorithm was traced to the inability of the algorithms to produce personalized recommendations, due to the fact that it solely on explicit ratings; therefore, the ratings matrix is considerably more sparse, and therefore in many cases the algorithm could not identify near neighbors or influencers and resorted to using "default" predictions (the average of all ratings for a venue in the database), which were not tailored to the tastes and likings of the individual users. The proposed algorithm tackles this shortcoming by exploiting textual information entered by the users to increase the ratings matrix density, achieving a considerable improvement in user satisfaction.

Within Fig. 5, we have also included user ratings for 10 individual recommendations (rec1-rec10); these have been chosen to demonstrate that even though the presented algorithm's performance is not uniform across all cases, it achieves the best results in the vast majority (97.3%) of all the cases (60users * 6 recommendations per algorithm * 3 recommendation algorithms).

This experiment clearly demonstrates that the inclusion of ratings generated on the basis of textual information entered in the form of SNs posts, in the recommendation process, as well as the use of appropriately assigned confidence metrics to those calculated ratings, proposed in this paper, provides a significant improvement in the quality of the generated recommendations, increasing the percentage of cases for which a recommendation can be generated, while maintaining similar accuracy levels to the cases that recommendations can be computed on the basis of explicitly entered ratings.

6 Conclusions and future work

In this paper, we have presented an algorithm which addresses the uncertainty inherent in the conversion of user textual reviews for venues, which are in abundance in social networks, to numeric ratings, by extracting and exploiting features of the review text. In particular, the algorithm uses the number of positive and negative terms in the document to compute a confidence metric for the ratings that are computed from texts, and subsequently uses this confidence metric to enhance the user similarity computation and rating prediction formulation process, elevating prediction accuracy. Subsequently, we have presented a venue recommendation algorithm that combines venue rating predictions with metrics about the QoS offered by the venues, as well as metrics about the similarity and physical distance between the venues, in order to generate personalized recommendations highly tailored to each user's personal profile.

The effectiveness of the proposed algorithms was evaluated regarding (i) rating prediction accuracy and (ii) social network users' satisfaction regarding the venue recommendations offered to them. The results are encouraging, introducing significant gains in social network user satisfaction. More specifically, the proposed algorithm has been found to raise user satisfaction by 24%, as compared to algorithms that rely only on explicitly entered ratings, while the improvement in terms of user satisfaction against algorithms that do not take into account the uncertainty inherent in textual review to rating conversions has been quantified to 8.3%.

Regarding our future work, we plan to test more word embedding models, such as (Bradley and Lang 1999; Brysbaert et al. 2014; Juhasz and Yap 2013) and we plan to conduct a user survey with a higher number of participants and more representative demographics.

Finally, we plan to extend our algorithm to consider information from the IoT (Margaris and Vassilakis 2017) and the type of company and the day/time of the visit or the place, since these aspects have been shown to play a significant role when rating places (Aliannejadi et al. 2017; Margaris and Vassilakis 2017); this will enable us to further normalize each particular rating, aiming for more accurate recommendations. The exploitation of images and multimedia information for deriving venue similarity (Wang et al. 2015) or user context will be also considered.

References

Aliannejadi M, Mele I, Crestani F (2016) User model enrichment for venue recommendation. In: Seffah A, Penzenstadler B, Alves C, Peng X (eds) Information retrieval technology. Lecture notes in computer science, vol 9994, pp 212–223

Aliannejadi M, Mele I, Crestani F (2017) Personalized ranking for context-aware venue suggestion. In: Proceedings of the symposium on applied computing, Marrakech, Morocco, pp 960–962

Aslam J, Montague M (2001) Models for metasearch. In: Proceedings of the 24th annual international conference on research and development in information retrieval, New Orleans, Louisiana, USA, pp 276–284

Bakshy E, Rosenn I, Marlow C, Adamic L (2012a) The role of social networks in information diffusion. In: Proceedings of the 21st international conference on World Wide Web, Lyon, France, pp 519–528

Bakshy E, Eckles D, Yan R, Rosenn I (2012b) Social influence in social advertising: evidence from field experiments. In: Proceedings of the 13th ACM conference on electronic commerce, Valencia, Spain, pp 146–161

Balabanovic M, Shoham Y (1997) Fab: content-based, collaborative recommendation. Commun ACM 40(3):66–72

Bradley MM, Lang PJ (1999) Affective norms for English words (ANEW): instruction manual and affective ratings. Technical Report C-1, The Center for Research in Psychophysiology, University of Florida

Brysbaert M, Warriner AB, Kuperman V (2014) Concreteness ratings for 40 thousand generally known English word lemmas. Behav Res Methods 46(3):904–911

Chen L, Chen G, Wang F (2015) Recommender systems based on user reviews: the state of the art. User Model User-Adap Inter 25(2):99–154

Chikersal P, Poria S, Cambria E, Gelbukh A, Siong CE (2015) Modelling public sentiment in Twitter: using linguistic patterns to enhance supervised learning. In: Gelbukh A (ed) Computational linguistics and intelligent text processing. Lecture notes in computer science, vol 9042, pp 49–65

Cieslik J (2017a) nyyelp: Predicting yelp review rating using recurrent neural networks. https://github.com/i008/nyyelp. Accessed 18 Nov 2018

Cieslik J (2017b) IPython notebook document for nyyelp: predicting yelp review rating using recurrent neural networks. https://github.com/i008/nyyelp/blob/master/nlp.ipynb. Accessed 31 Nov 2018

Cocarascu O, Toni F (2018) Combining deep learning and argumentative reasoning for the analysis of social media textual content using small data sets. Comput Linguist 44(4):833–858

Contratres FG, Alves-Souza SN, Filgueiras LVL, DeSouza LS (2018) Sentiment analysis of social network data for cold-start relief in recommender systems. In: WorldCIST'18 2018. Proceedings of the 6th world conference on information systems and technologies, Naples, Italy, pp 122–132

Eirinaki M, Gao J, Varlamis I, Tserpes K (2018) Recommender systems for large-scale social networks: a review of challenges and solutions. Future Gener Comput Syst 78(1):413–418

Fan M, Khademi M (2014) Predicting a business star in yelp from its reviews text alone. arXiv preprint arXiv:1401.0864. Accessed 15 Nov 2018

Gilbert E, Karahalios K (2009) Predicting tie strength with social media. In: Proceedings of the SIGCHI conference on human factors in computing systems, Boston, USA, pp 211–220

Gregory B (2013) Kaggle YELP business rating prediction. https://github.com/theusual/kaggle-yelp-business-rating-prediction. Accessed 22 Nov 2018

He D, Wu D (2008) Toward a robust data fusion for document retrieval. In: Proceedings of the 2008 international conference on natural language processing and knowledge engineering (NLP-KE '08). https://doi.org/10.1109/nlpke.2008.4906754

Herlocker JL, Konstan JA, Terveen LG, Riedl JT (2004) Evaluating collaborative filtering recommender systems. ACM Trans Inf Syst 22(1):5–53

ITU (1998) Recommendation E.800 Quality of service and dependability vocabulary. Blue Book, Fascicle II.3

Jameel S, Liao Y, Lam W, Schockaert S, Xie X (2016) Exploring urban lifestyles using a nonparametric temporal graphical model. In: Proceedings of the 2016 ACM international conference on the theory of information retrieval, Newark, DE, USA, pp 251–260

Jones CB, Alani H, Tudhope D (2001) Geographical information retrieval with ontologies of place. In: Proceedings of the 2001 conference on spatial information theory, Morro Bay, CA, USA, pp 322–335

Juhasz BJ, Yap MJ (2013) Sensory experience ratings for over 5000 mono- and disyllabic words. Behav Res Methods 45(1):160–168

Keras (2018) The python deep learning library. https://keras.io/. Accessed 31 Nov 2018

Li Y, Nie J, Zhang Y, Wang B, Yan B, Weng F (2010) Contextual recommendation based on text mining. In: Proceedings of the 23rd international conference on computational linguistics, Beijing, China, pp 692–700

Liu B (2017) Opinion lexicon (or sentiment lexicon). https://www.cs.uic.edu/~liub/FBS/sentiment-analysis.html. Accessed 24 Nov 2018

Liu F, Lee HJ (2010) Use of social network information to enhance collaborative filtering performance. Expert Syst Appl 37:4772–4778. https://doi.org/10.1016/j.eswa.2009.12.061

Liu B, Hu M, Cheng J (2005) Opinion observer: analyzing and comparing opinions on the web. In: Proceedings of the 14th international World Wide Web conference, Chiba, Japan, pp 342–351

Maks I, Izquierdo R, Frontini F, Agerri R, Azpeitia A, Vossen P (2014) Generating Polarity Lexicons with WordNet propagation in five languages. In: Proceedings of the 9th international conference on language resources and evaluation, Reykjavik, Iceland, pp 1156–1161

Margaris D, Vassilakis C (2017) Exploiting Internet of Things information to enhance venues' recommendation accuracy. Serv Oriented Comput Appl 11(4):393–409

Margaris D, Vassilakis C (2018a) Dataset for rating prediction for social media. https://github.com/costasvassilakis/socialMediaRatingPrediction. Accessed 2 March 2019

Margaris D, Vassilakis C (2018b) Exploiting rating abstention intervals for addressing concept drift in social network recommender systems. Information 10(7), 230

Margaris D, Vassilakis C (2018c) Improving collaborative filtering's rating prediction accuracy by considering users' rating variability. In: Proceedings of the 4th IEEE international conference on big data intelligence and computing, Athens, Greece, pp 1022–1027

Margaris D, Vassilakis C, Georgiadis P (2015) An integrated framework for adapting WS-BPEL scenario execution using QoS and collaborative filtering techniques. Sci Comput Program 98:707–734

Margaris D, Vassilakis C, Georgiadis P (2016) Recommendation information diffusion in social networks considering user influence and semantics. Soc Netw Anal Min 6(108):1–22

Margaris D, Vassilakis C, Georgiadis P (2017) Knowledge-based leisure time recommendations in social networks. Current trends on knowledge-based systems: theory and applications. Springer, Berlin, pp 24–48

Margaris D, Vassilakis C, Georgiadis P (2018) Query personalization using social network information and collaborative filtering techniques. Future Gener Comput Syst 78(1):440–450

Maurya CG, Gore S, Rajput DS (2018) A use of social media for opinion mining: an overview (with the use of hybrid textual and visual sentiment ontology). In: Proceedings of international conference on recent advancement on computer and communication, Bhopal, India, pp 315–324

McAuley J, Leskovec J (2013) Hidden factors and hidden topics: understanding rating dimensions with review text. In: Proceedings of the 7th ACM international conference on recommender systems, Hong Kong, China, pp 165–172

Mersha T, Adlakha V (1992) Attributes of service quality: the consumers' perspective. Int J Serv Ind Manag 3(3):34–45. https://doi.org/10.1108/09564239210015157

Monfil-Contreras EU, Alor-Hernández G, Cortes-Robles G, Rodriguez-Gonzalez A, Gonzalez-Carrasco I (2013) RESYGEN: a recommendation system generator using domain-based heuristics. Expert Syst Appl 40(1):242–256

Moshfeghi Y, Piwowarski B, Jose JM (2010) Handling data sparsity in collaborative filtering using emotion and semantic based features. In: Proceedings of the 34th international ACM SIGIR conference on research and development in information retrieval, Beijing, China, pp 625–634

Musat CC, Liang Y, Faltings B (2013) Recommendation using textual opinions. In: Proceedings of the 23rd international joint conference on artificial intelligence, Beijing China, pp 2684–2690

Pang B, Lee L, Vaithyanathan S (2002) Thumbs up? Sentiment classification using machine learning techniques. In: Proceedings of the 2002 conference on empirical methods in natural language processing, Stroudsburg, PA, pp 79–86

Pennington J, Socher R, Manning CD (2014) GloVe: global vectors for word representation. In: Proceedings of the conference on empirical methods in natural language processing, Doha, Qatar, pp 1532–1543

Pero Š, Horváth T (2013) Opinion-driven matrix factorization for rating prediction. In: Proceedings of the 21st international conference on user modeling, adaptation and personalization, Rome, Italy, pp 1–13

Pirasteh P, Jung JJ, Hwang D (2014) Item-based collaborative filtering with attribute correlation: a case study on movie recommendation. In: Proceedings of the 6th Asian conference on intelligent information and database systems (ACIIDS 2014) Bangkok, Thailand, 7–9 April 2014, Proceedings, Part II, pp 245–252

Poirier D, Fessant F, Tellier I (2010) Reducing the cold-start problem in content recommendation through opinion classification. In: Proceedings of the 2010 IEEE/WIC/ACM international conference on web intelligence and intelligent agent technology, Toronto, Canada, pp 204–207

Raghavan S, Gunasekar S, Ghosh J (2012) Review quality aware collaborative filtering. In: Proceedings of the 6th ACM conference on recommender systems, Dublin, Ireland, pp 123–130

Ritter A, Clark S, Etzioni M, Etzioni O (2011) Named entity recognition in tweets: an experimental study. In: Proceedings of the 2011 conference on empirical methods in natural language processing, Edinburgh, United Kingdom, pp 1524–1534

Seroussi Y, Bohnert F, Zukerman I (2011) Personalised rating prediction for new users using latent factor models. In: Proceedings of the 22nd ACM conference on hypertext and hypermedia, Eindhoven, The Netherlands, pp 47–56

Shardanand U, Maes P (1995) Social information filtering: algorithms for automating "Word of Mouth''. In: Proceedings of the 1995 SIGCHI conference on human factors in computing systems, Denver, Colorado, USA, pp 210–217

Tiago C, Soares C, Carvalho A (2018) Metalearning and recommender systems: a literature review and empirical study on the algorithm selection problem for collaborative filtering. Inf Sci 423:128–144

Turney PD (2002) 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, Stroudsburg, PA, USA, pp 417–424

Van der Maaten LJP (2014) Accelerating t-SNE using tree-based algorithms. J Mach Learn Res 15:3221–3245

Wang X, Zhao Y-L, Nie L, Gao Y, Nie W, Zha Z-J, Chua T-S (2015) Semantic-based location recommendation with multimodal venue semantics. IEEE Trans Multimed 17(3):409–419

Yang P, Fang H (2015) Combining opinion profile modeling with complex context filtering for contextual suggestion. In: Proceedings of the 24th text retreival conference, Gaithersburg, USA, pp 1–4

Yun Y, Hooshyar D, Jo J, Lim H (2018) Developing a hybrid collaborative filtering recommendation system with opinion mining on purchase review. J Inf Sci 44(3):331–344

Zhao G, Qian X, Xie X (2016) User-service rating prediction by exploring social users' rating behaviors. IEEE Trans Multimed 18(3):496–506

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.