A User-Centered Approach for Social Recommendations

F. Colace, M. De Santo and L. Greco

Dipartimento di Ingegneria dell'Informazione,
Ingegneria Elettrica e Matematica Applicata
University of Salerno
Fisciano, Italy
Email: {fcolace,desanto,lgreco}@unisa.it

F. Amato, V. Moscato, F. Persia and A. Picariello

Dipartimento di Ingegneria Elettrica e
Tecnologie dell'Informazione
University of Naples
Naples, Italy
Email: {flora.amato,vmoscato,fabio.persia,picus}@unina.it

***Abstract*—Recommender Systems represent useful tools helping users to find "what they need" from a very large number of candidates and supporting people in making decisions in several contexts. In this paper, we propose a novel user-centered and social recommendation approach in which several aspects related to users, i.e., *preferences*, *opinions*, *behavior*, *feedbacks*, are considered and integrated together with *items' features* and *context information* within a general framework that can support different applications using proper customizations (e.g., recommendation of news, photos, movies, travels, etc.). Preliminary experiments on system accuracy show how our approach provides very promising and interesting results.**

Keywords*–*Recommender Systems; Sentiment Analysis; Context-Awareness.

I. Introduction

In the era of *Big Data*, we are assisting to an explosive and amazing increase of digital information and, as a consequence, more and more huge data collections of different nature are widely available to a large population of users. In addition, the widespread diffusion of the most popular social networks, multimedia repositories, news archives, travel websites, e-commerce portals, and so on, has constrained users necessarily to deal with this ocean of information to find "what they need".

In the last decade, *Recommender Systems* have been introduced to facilitate the browsing of such collections leveraging several features to provide useful recommendations: user preferences and past behavior, preferences and past behavior of the user community, items' features and how they can match user preferences, user feedbacks, context information and how recommendations can change together with the context. To accomplish their goals, the last generation of recommender systems is usually composed by one or more of the following components [1]: (i) a *pre-filtering* module that selects for each user a set of objects that are good candidates to be recommended (such objects usually match user preferences and needs or satisfy context contraints); (ii) a *ranking* module that assigns to every user a rank related to each candidate object using recommendation techniques (that can exploit in several ways items' features and users' preferences, feedbacks and behaviors); (iii) *post-filtering* module that dynamically excludes, for each user, some items from the recommendations' list on the base of the collected feedbacks and context information.

In this paper, we propose a novel *user-centered* approach that provides *social* recommendations, capturing and exploiting several aspects related to users: *preferences* (usually coded in the shape of items' metadata), *opinions* (textual comments to which it is possible to associate a particular sentiment), *behavior* (in most cases, logs of past items' observations made by users), *feedbacks* (usually expressed in the form of ratings). All these features are considered and integrated together with *items' features* and *context information* within a general and unique recommendation framework that can support different applications using proper customizations (e.g., recommendation of news, photos, movies, travels, etc.), overcoming problems related to the availability and quality of user profiles and ratings.

As motivating example, we can consider a user who desires to have information about the coming soon movies. In this case, the list of suggested items should consider the following information: (i) user preferences in terms of movies' metadata (e.g., favorite genre, director, stars, etc.); (ii) item features (i.e., movies' metadata) and their similarity (e.g., a semantic relatedness based on a movie taxonomy); (iii) user behavior in terms of the sequence of items that in the past the community of users have observed and positively rated; (iv) user feedbacks in terms of the user community ratings; (v) user opinions in terms of the average sentiment that items have aroused on the user community; (vi) context information: in this case, in terms of item features that satisfy the search criteria (e.g., coming soon movies shown in theaters near the user) or that have a good similarity with respect to the item that the user has selected and is currently watching.

For instance, we can imagine a user who prefers the adventure and fantasy genres and who has among his/her favorite actors Ian McKellen and Hugh Jackman; the system can initially suggest as first items to watch the X-Men saga movies, together with other titles. After the pre-filtering stage, the candidate items matching user preferences are initially ranked on the base of the related social *popularity* (e.g., number of accesses through past users' paths, average rating and sentiment from users' reviews, etc.). Successively, if the user selects to read more information about one of X-Men movies and rates it positively, a list of filtered items is then provided with the most popular movies that have a certain similarity in terms of metadata with the X-Men movie.

 ISBN: 978-1-61208-382-7

Eventually, if the user chooses to limit the search to the coming soon movies (constraint on a metadata value) and selects his/her position as context information, all the best movies - in according to a social view - matching user preferences that are showing in the next days in theaters near the user will be finally proposed (e.g., "X-Men: Days of Future Past" and "Captain America: The Winter Soldier").

The paper is organized as follows. Section 2 discusses the state of the art of other similar systems, while Section 3 describes the proposed strategy for recommendation. Section 4 illustrates a system customization in the domain of movie recommendation and reports some preliminary experimental results, and provides a comparison with other recommendation techniques. Finally, Section 5 gives some concluding remarks and discusses future work.

II. Related Works

Recommender Systems represent a meaningful response to the problem of information overload since the mid-1990s [2], when the early works on this topic have been proposed. The main aim is to predict user's preferences and make meaningful suggestions about items that could be of interest [1].

In *content-based* approach, the system recommends an item to a user relying on the ratings made by the user himself for *similiar* items in the past. The similiarity between items is often computed by the use of information retrieval and filtering techniques [3]. However, a critical drawback of this approach is *overspecialization*, since the systems only recommend items similiar to those already rated by the user. In *collaborative filtering* [4], the recommendation is performed by filtering and evaluating items with respect to ratings from other users. Typically, users are asked to rate items and a similarity between their profiles is also computed to be used as a weight when making recommendations for highly rated items [5]. An important limitation of collaborative filtering systems is the *cold start problem*, that describes situations in which a recommender is unable to make meaningful recommendations due to an initial lack of ratings, thus degrading the filtering performance. Content-based filtering and collaborative filtering may be combined in the so called *hybrid* approach that helps to overcome limitations of each method [6]. A recommendation strategy eventually should be also able to provide users with the more relevant information depending on the *context* [7][8][9] (i.e., user preferences, user location, observed objects, weather and environmental conditions, etc.) as in *Context Aware Recommendation Systems*.

Performance of recommender systems is strictly related to the *availability* and *quality* of user profiles and ratings and an important improvement to overcome such problems lies in the possibility to embed *social* elements into a recommendation strategy [10]. In such a context, customer *opinion summarization* [11] and *sentiment analysis* [12] techniques represent effective augmentations to traditional recommendation strategies, for example by not recommending items that receive a lot of negative feedbacks [10]. Finally, a recent category of recommenders, named *Large Scale Recommender Systems* (LSRS) [13], calls for new capabilities of such applications to deal with very large amount of data with respect to scalability and efficiency issues. Distributed computing of recommendations and parallel *matrix factorization* are the most diffused approaches to cope with such a problem.

Our work exploits sentiment classification techniques based on the *Latent Dirichlet Allocation* (LDA) to refine and enrich a context-aware and hybrid recommendation strategy that some of the authors have proposed for recommendation in multimedia browsing systems [14][15][16]. We thus obtained a user-centered approach in which several aspect of a user (preferences, opinions, feedbacks, behaviors) are simultaneously considered together with item features and context information within a unique and general framework able to efficiently scale with the increase of data.

III. The Recommendation Strategy

The basic idea behind our proposal is that when a user is browsing a particular items' collection, the recommender system: (i) determines a set of useful *candidate* items for the recommendation, on the base of user actual needs and preferences (*pre-filtering stage*); (ii) opportunely assigns to these items a rank, previously computed exploiting items' intrinsic features and users' past behaviors, and using as refinement, social information in the shape of users' opinions and feedbacks (*ranking stage*); (iii) dynamically, when a user "selects" as interesting one or more of the candidate objects, determines the list of the most suitable items (*post-filtering stage*), also considering other context information expressed by users in the shape of constraints on items' features.

A. Pre-filtering Stage using user preferences

In the *pre-filtering* stage, our aim is to select for a given user u_h a subset $O_h^c \subset O$ (O being the set of items) containing items that are good "candidates" to be recommended: such items usually have to match some (static) user preferences and (dynamic) actual needs. Each item subject to recommendation may be represented in different and heterogeneous feature spaces and the first step consists in clustering together "similar" items, where the similarity should consider all (or subsets of) the different spaces of features. To this purpose, we employ *high-order star-structured co-clustering* techniques - that some of the authors have adopted in a previous work [16] - to address the problem of heterogeneous data pre-filtering. The pre-filtering stage leverages the clustering results to select a set of items by using the user's profile, which is modeled as sets of descriptors in the same spaces as the items' descriptors.

B. Ranking Stage using user behavior and items similarity

We use a technique that some of the authors have proposed in previous works - combining low and high level features of items, past behavior of individual users and overall behavior of the whole user "community" [14][15] - to provide useful recommendations during the browsing of multimedia collections. Our basic idea is to assume that when an item o_i is chosen after an item o_j in the same user *browsing session* (and both the explored items have been positively rated or have captured attention of users for an adequate time), this event means that o_i "is voting" for o_j. Similarly, the fact that an item o_i is "very similar" in terms of some intrinsic features to o_j can also be interpreted as o_j "recommending" o_i (and viceversa). Thus, we are able to model a browsing system for the set of items O as a labeled graph (coding both browsing sessions of the different users and similarity between items' pairs by means of a set of proper matrices), and to compute the *recommendation grade* $\rho(o_i)$ for an item $o_i \in O_h^c$ related to a given user u_h in a similar manner to *Google Page Rank* algortihm [15].

 ISBN: 978-1-61208-382-7

C. Refining items ranks using user sentiments and feedbacks

We used the sentiment extraction technique as an improvement of the approach presented by some of the authors in a previous work [17], where the LDA has been adopted for mining the sentiment inside documents. In our view, the knowledge within a set of documents can be represented in a compact fashion by the use of a complex structure - the *mixed Graph of Terms* (mGT) - that contains the most discriminative words and the probabilistic links between them. The mGT is built starting from a set of comments belonging to a well-defined knowledge domain and manually labeled according to the sentiment expressed within them. In this way, the mGT contains words (and their probabilistic relationships) which are representative of a certain sentiment for that knowledge domain.

For the ranking refinement, we introduce two probabilities P^+ and P^- which express the probability that a sentiment, extracted from the set of comments related to a given item, is "positive" or "negative" (the probabilities P^+ and P^- also take into account the overall rating and trustiness of users). Such probabilities are then combined with the overall rank of an item by a proper function that increases the recommendation grade value if the sentiment within item's comments is positive, in the opposite decreases it in the case of negative mood.

D. Post-Filtering Stage using context information

In this stage, we have introduced a *post-filtering* method for generating the final set of "real" candidates for recommendation. The set of candidates includes the items that have been accessed by at least one user within k steps from a selected object o_j and the items that are most similar to o_j according to the results of a *Nearest Neighbor Query* ($NNQ(o_j, O_h^c)$) functionality.

The ranked list of recommendations is then generated by ranking the candidates set for each object o_j selected as interesting by user u_h. Finally, for each user, all the items that do not respect possible context constraints are removed from the final list. In our model, *context constraints* are expressed in terms of assigned values to the elements of particular sub-classes of features that the recommended items have to satisfy.

IV. Preliminary Experimental Results

A. Using the system for recommending movies

We have opportunely customized our system in order to provide recommendation services for users that are interested in coming soon movies. The design choices are briefly reported in the following: (i) we consider as data source the *IMDB* web site, collecting about 10,000 items; (ii) as items' metadata, we consider for each movie information related on *title*, *genre*, *stars*, *description*, *year*, *director* and *list of theaters* (characterized by name and location) in which they are coming; (iii) for each movie, available users' preferences, comments and feedbacks have been captured, also exploiting correlated public information from Social Networks (i.e., Facebook); (iv) users' behaviors have been reconstructed considering the available log with time-stamped information of users that have positively rated or watching for a certain time some items in the same browsing session.

B. Accuracy Computation

We decided to perform for the movie recommendation problem an evaluation based on *accuracy* metrics [18]. We used the dataset provided by [19], which makes available data collected by the *MovieLens* [20] recommender system. Through its website, MovieLens collects the preferences expressed by a community of registered users on a huge set of movie titles. The adopted dataset contains (i) explicit ratings about 1682 movies made by 943 users, (ii) demographic information about users (age, gender, occupation, zip code), and (iii) a brief description of the movies (title, release year, genres). We then determined user preferences by considering the known preferences of similar users (e.g., same age, occupation, etc.) from social networks and extended items' features by considering IMDB metadata, as well as, we exploited IMDB users' comments.

The experiments have been conducted on a collection of about 1,000 movies, rated by a subset of 100 users: each of them had rated at least 150 movies and at most 300, assigning to each movie a score between 1 (*"Awful"*) and 5 (*"Must see"*). Additionally, using the *timestamp* information, we were able to reconstruct usage patterns for each user and consequently the browsing matrices.

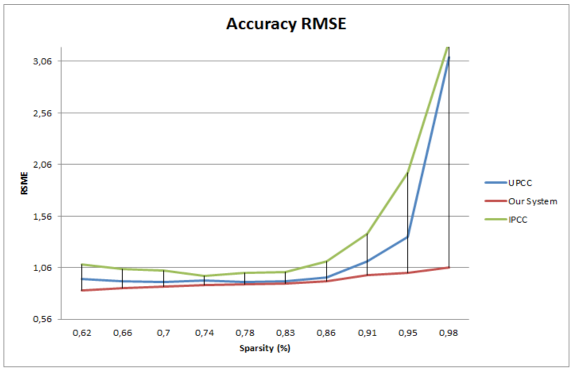
We used the *Mean Absolute Error* (*MAE*) and the *Root Mean Square Error* (*RMSE*) as metrics in our experiments. In our case, *MAE* and *RMSE* are defined as: $MAE = \frac{1}{N}\sum_{u,i,j}|r_{ui}^j - \hat{r}_{ui}^j|$ and $RMSE = \sqrt{\frac{1}{N}\sum_{u,i,j}(r_{ui}^j - \hat{r}_{ui}^j)^2}$; where r_{ui}^j is the actual rating that the user u has given to item i for the item j, $\hat{r}_{ui}^j$ is the system predicted rating, and N is the total number of test ratings.

We compared the achieved accuracy of the predictions computed by our recommender system with the *UPCC* and *IPCC* [5] approaches (which reliable implementation can be obtained leveraging machine learning libraries provided by the *Apache Mahout* framework).

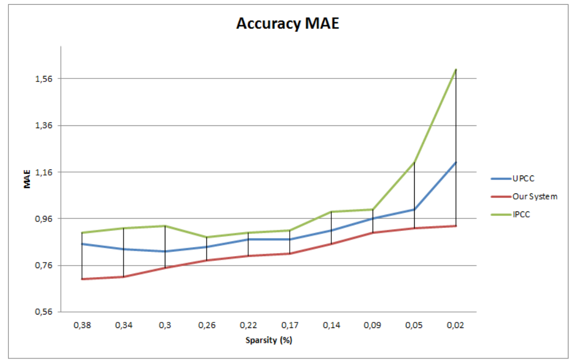
Fig. 1 shows the trend of RMSE and MAE for our system as well as for the UPCC and IPCC algorithms, as the sparsity of the rating matrix increases. Our approach outperforms UPPC and IPPC ones for each value of items' sparsity - and especially for higher values - showing how social information can improve recommendations. This is also due to the use of the items' *similarity matrix*, which provides useful information to the algorithm, in order to compute meaningful predictions even if a user's browsing session data is not available. Thus, our approach does not suffer from the *cold start problem*.

C. Considerations on efficiency

In order to evaluate the efficiency of our recommender system, we have measured execution times w.r.t. the execution times of other state-of-the-art methods. As the recommendation grades computation can be performed in off-line manner and the related updates are correlated to the insertion of a new item (or an update of its features) or to a new user, the running time is essentially dependent on the size of candidate items' set obtained in the pre-filtering stage. In general, we observed that the average computation times for all methods are comparable and it takes at most few seconds to obtain useful recommendations also for large sets of candidates.



(a) RMSE comparison



(b) MAE comparison

Figure 1. Comparison in terms of MAE and RMSE between our approach and UPCC and IPCC respectively

For scalability issues, we decided to use a distributed approach using *Apache Hadoop* framework to compute the items' clusters in the pre-filtering stage.

V. Conclusions and Future Work

We described a user-centered and social recommendation approach in which several aspects related to users - i.e., *preferences*, *opinions*, *behavior*, *feedbacks* - are considered and integrated together with *items' features* and *context information* within a general framework. We focused on a particular case study and implemented a recommender system based on our innovative approach, which is able to help users to choose coming soon movies having IMDB as main data source. Then, we investigated the effectiveness of the proposed approach in the considered scenarios, through the evaluation of the *accuracy*. Summing up, future efforts will be devoted to (i) extending the experimental evaluation to larger datasets, also considering the *stability* of recommendations, (ii) applying our approach to other kinds of data from heterogeneous collections and comparing it with other recent approaches of the literature.

References

[1] F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, Eds., Recommender Systems Handbook. Springer, 2011.

[2] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl, "Grouplens: An open architecture for collaborative filtering of netnews." ACM Press, 1994, pp. 175–186.

[3] M. Pazzani and D. Billsus, "Content-Based Recommendation Systems," 2007, pp. 325–341. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-72079-9_10

[4] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions," IEEE Transactions on Knowledge and Data Engineering, vol. 17, 2005, pp. 734–749.

[5] X. Su and T. Khoshgoftaar, "A survey of collaborative filtering techniques," Advances in Artificial Intelligence, vol. 2009, 2009.

[6] A. I. Schein, A. Popescul, L. H. Ungar, and D. M. Pennock, "Methods and metrics for cold-start recommendations," in Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, ser. SIGIR '02. New York, NY, USA: ACM, 2002, pp. 253–260. [Online]. Available: http://doi.acm.org/10.1145/564376.564421

[7] P. Dourish, "What we talk about when we talk about context," Personal and ubiquitous computing, vol. 8, no. 1, 2004, pp. 19–30.

[8] K. Kabassi, "Personalisation systems for cultural tourism," in Multimedia services in intelligent environments. Springer, 2013, pp. 101–111.

[9] A. Karatzoglou, X. Amatriain, L. Baltrunas, and N. Oliver, "Multiverse recommendation: n-dimensional tensor factorization for context-aware collaborative filtering," in Proceedings of the fourth ACM conference on Recommender systems. ACM, 2010, pp. 79–86.

[10] X. Zhou, Y. Xu, Y. Li, A. Josang, and C. Cox, "The state-of-the-art in personalized recommender systems for social networking," Artif. Intell. Rev., vol. 37, no. 2, Feb. 2012, pp. 119–132. [Online]. Available: http://dx.doi.org/10.1007/s10462-011-9222-1

[11] L. Zhuang, F. Jing, and X.-Y. Zhu, "Movie review mining and summarization," in Proceedings of the 15th ACM International Conference on Information and Knowledge Management, ser. CIKM '06. New York, NY, USA: ACM, 2006, pp. 43–50. [Online]. Available: http://doi.acm.org/10.1145/1183614.1183625

[12] X. Ding, B. Liu, and P. S. Yu, "A holistic lexicon-based approach to opinion mining," in Proceedings of the 2008 International Conference on Web Search and Data Mining, ser. WSDM '08. New York, NY, USA: ACM, 2008, pp. 231–240. [Online]. Available: http://doi.acm.org/10.1145/1341531.1341561

[13] H.-F. Yu, C.-J. Hsieh, S. Si, and I. S. Dhillon, "Parallel matrix factorization for recommender systems," Knowledge and Information Systems, 2013, pp. 1–27.

[14] M. Albanese, A. d'Acierno, V. Moscato, F. Persia, and A. Picariello, "A multimedia semantic recommender system for cultural heritage applications," in Semantic Computing (ICSC), 2011 Fifth IEEE International Conference on. IEEE, 2011, pp. 403–410.

[15] M. Albanese, A. dAcierno, V. Moscato, F. Persia, and A. Picariello, "A multimedia recommender system," ACM Transactions on Internet Technology (TOIT), vol. 13, no. 1, 2013, p. 3.

[16] I. Bartolini, V. Moscato, R. G. Pensa, A. Penta, A. Picariello, C. Sansone, and M. L. Sapino, "Recommending multimedia visiting paths in cultural heritage applications," Multimedia Tools and Applications, 2014, pp. 1–30.

[17] F. Colace, M. De Santo, and L. Greco, "A probabilistic approach to tweets' sentiment classification," in Affective Computing and Intelligent Interaction (ACII), 2013 Humaine Association Conference on, 2013, pp. 37–42.

[18] G. Adomavicius and J. Zhang, "Stability of recommendation algorithms," ACM Transactions on Information Systems (TOIS), vol. 30, no. 4, 2012, p. 23.

[19] "http://www.grouplens.org."

[20] "http://movielens.umn.edu."