

# A Hybrid Recommender System: User Profiling from Keywords and Ratings

Ana Stanescu, Swapnil Nagar, Doina Caragea  
Kansas State University  
Manhattan KS, USA  
anas@ksu.edu, zealswap@ksu.edu, dcaragea@ksu.edu

**Abstract**—Over the last decade, user-generated content has grown continuously. Recommender systems that exploit user feedback are widely used in e-commerce and quite necessary for business enhancement. To make use of such user feedback, we propose a new content/collaborative hybrid approach, which is built on top of the recently released hetrec2011-movielens-2k dataset and is an extension of a previously proposed neighborhood based approach, called Weighted Tag Recommender (WTR). Our approach has two versions. Both versions make use of ratings to enable collaborative filtering and use either user tags, available in the hetrec2011-movielens-2k dataset, or movie keywords retrieved from IMDB, to capture movie content information. Experimental results show that the information from keywords can help build a movie recommender system competitive with other neighborhood based approaches and even with more sophisticated state-of-the-art approaches.

**Keywords**-Recommender systems; keywords; tags; ratings.

## I. INTRODUCTION

As Web 2.0 applications continue to proliferate, the overabundant unstructured data that becomes available on the Internet contains great amounts of useful knowledge. Overwhelmed by the huge number of options presented with, people rely more and more on other peoples' previous experiences for choosing movies, books and other products.

The collective intelligence is allegedly better than the experts at making recommendations [21]. Surowiecki's *wisdom of crowds* (WOC) hypothesis states that when it comes to popular culture, among other domains, averaging the opinions of a large group of people captures the reality more accurately than legitimate experts. Such opinions are expressed online, and they constitute a great source of raw data that Recommender Systems (RS) [17] can process in order to make suggestions for those who are seeking them. Although there are several factors that influence the quality of such aggregations [13], recommendations based on common viewpoints become more and more trustworthy and widely-used [19]. The recommendation task is often-times reduced to the problem of estimating what rating a user would give for an unseen item, or to finding a list of items that the user is most likely to enjoy.

Generally, recommendation systems can be categorized as content-based, collaborative or hybrid [3], as described below:

- **Content-Based** (CB) recommender systems suggest items similar to the ones that the user preferred in

the past. However, there are some limitations to the CB technique, like the data scarcity problem. Modeling the user's interest is limited to extracting features from their history [3]. Another limitation is that CB systems cannot identify new and different items that the user may enjoy, as it is prone to finding only those that are highly similar to the items in the history of that user.

- **Collaborative Filtering** (CF) systems [20] filter large data sets in search for patterns and information of interest, by collecting preferences from multiple users. Collaborative recommendations are based on the user-user similarity. The system will recommend those items that are liked by users with matching taste. Similar to the CB approach, CF techniques have shortcomings as well: a new item cannot be recommended until someone references it. Moreover, a user with unusual preferences may not receive recommendations unless there are other users that exhibit the same interests.
- A combination of the CB and CF techniques, referred to as **hybrid** approaches, can alleviate some of the problems that the CB and CF systems encounter individually, and can generally achieve superior results.

In this work, we propose a hybrid system for movie recommendations that mediates the data sparsity problem and reduces the noise from the user generated content.

We adapt for movies the Weighted Tag Recommender (WTR) approach from [14] – where the authors addressed the problem of recommending books on Amazon and built their system exclusively from tag information. The WTR approach can utilize the tag information available in the hetrec2011-movielens-2k dataset, but it does not use explicit ratings. As opposed to WTR, our modified approach can make use of ratings to capture collaborative filtering and user-tags to capture movie content information. We refer to this variant of our approach as the Weighted Tag-Rating Recommender (WTRR).

Tags provide specific information for a user, but limit the usage of the data to the user-movie pairs that have tags (significantly smaller number compared with all pairs that have ratings). To mediate this issue, we propose a second variant of our approach called Weighted Keyword-Rating Recommender (WKRR). This variant uses ratings, along with movie keywords retrieved from IMDB, instead of tags. Movie keywords (which are not user specific) allow us to use

all ratings available in hetrec2011-movielens-2k, as WKRR associates the content information from movies with the users, based on their ratings.

Both our keyword and tag representations of users can help alleviate the noise and semantic ambiguity problems inherent in the information contributed by users of social networks. Experiments on a subset of the dataset (which contains both ratings and tags) show that the WTRR approach slightly outperforms the WKRR approach. However, WKRR can be applied to the whole hetrec2011-movielens-2k dataset and results show that the information from keywords can help build a movie recommender system comparable with other neighborhood based approaches and even with more sophisticated state-of-the-art approaches.

Given this background, the rest of the paper is organized as follows: We discuss related work in Section II. We give a detailed explanation of our approach in Section III. Following that, we discuss the experimental design in Section IV and show the results in Section V. We conclude our study in Section VI.

## II. RELATED WORK

The work on recommender systems has been expanding greatly and the results are constantly improving. Recommender systems are currently applied to a wide range of domains, from entertainment to scholarly articles, from products to friend suggestions in social networks [1]. One culminating point which attracted a lot of interest in recommender systems and also attention from the media was the announcement of the million-dollar prize from Netflix [5], which required a 10% improvement over Netflix's best recommendation technique at that time.

For comprehensive information on recommender systems, the reader is referred to [16], [4]. In what follows, we will review some prior studies that are using tags in the process of making recommendations. We will also look at other papers that have used the hetrec2011-movielens-2k dataset, as they are the most relevant to our work.

Tagging is a type of labeling, whose purpose is to assist users in the process of finding content on the web. It has evolved considerably thanks to social networks and has become a very popular concept. In 2004, Thomas Vander Wal assigned the name "folksonomy"<sup>1</sup> to the tag system developed by Web 2.0 consumers, as a derivation from the phrase "people's taxonomy." Although the tagging terms are highly personalized, their aggregation conveys a sound basis for prediction algorithms. For example, Said [18] proposes a folksonomy-based approach to personalize tags. For each user, each tag is assigned a value obtained from averaging the ratings the user gave to the movies tagged by that particular tag.

Tags are free annotations and there are no constraints enforced when it comes to assigning tags. This makes tag-based recommender systems suffer from degraded performance because of semantic problems, such as polysemy and synonymy [10]. A hybrid system proposed by Liang et al. [14] addresses these problems, by using weighted tags, and was developed to recommend books from the Amazon database. The authors manage to overcome the problem of personal tagging by extending the pool of user tags to related tags that represent similar topics. Although the approach proved to be effective in improving the predictions, it has some limitations. The main limitation is that, while building tag and item profiles for users, the algorithm does not consider explicit ratings, but implicit ratings: if a user has tagged an item, then it is inferred that the user is interested in that item. However, that may not always be the case. Users may still tag movies that they do not like, in which case, the rating holds the information about their true preference. For domains like movies, books or any products where both tags and ratings are available, a recommender system should exploit all the information and it should not ignore the ratings. Systems that leverage ratings, which can be either explicitly provided by the users or implicitly inferred by the system, are known to perform well, e.g. Netflix [5]. However, ratings can also be noisy [2], therefore, in our approach we combine them with features corresponding to movie descriptions, such as tags or keywords.

The following approaches make use of the hetrec2011-movielens-2k dataset, which we also use in this paper:

The system proposed by [6] is an ensemble of various recommenders, called Information Market Based recommender fusion (IMBrf), primarily used for mining and aggregating the information from various sources. This technique is inspired from the market, where information from heterogeneous sources is incorporated to make predictions about future events. We compare our results with the results of other approaches as reported in [6], including a pure collaborative filtering technique (CF) and a content-based recommender system, called content analysis (CA). For CF, the authors used the neighborhood based approach and set the size of the neighborhood to 30. We will preserve these settings in our experimental setup, to be able to make fair comparisons. The CA recommender is based on latent topic analysis, and movies are mapped to topics via tags. The prediction is made by finding topics in new movies that are correlated to the user profiles.

Another recommender based on averaging the ratings (AVGR) is presented in [6], where an unrated item's rating is estimated from the weighted average of other ratings from other users. Finally, Linear Least Square (LLS) is proposed in [6]. LLS is a linear combination of CF, CA and AVGR:  $R_{LS} = \alpha CF + \beta CA + \gamma AVGR$ , where  $R_{LS}$  denotes the predicted rating (the parameters  $\alpha$ ,  $\beta$  and  $\gamma$  are found by optimization, for more details see [6]).

<sup>1</sup><http://vanderwal.net/folksonomy.html>

In [12], the authors propose learning multiple models which can incorporate different types of inputs to predict the preferences of diverse users. Probabilistic Matrix Factorization (PMF) is a variational Bayesian inference technique, used to alleviate the over-fitting problem in singular value decomposition (SVD) approaches. Priors are introduced and parameters estimated using variational Bayesian inference [15]. PMF models the user preference matrix as a product between the lower-rank user and movie matrices [12].

### III. APPROACHES

We provide an overview of our proposed Weighted Tag-Rating Recommender variant in Section III-A. The Weighted Keyword-Rating Recommender variant is described in Section III-B. We discuss how the neighborhood is formed in Section III-C, and finally, in Section III-D we define the prediction scheme we used.

#### A. WTRR

The book recommender system proposed in [14] is built from tag information only. The authors state that tags are sufficient for capturing the content information of items. However, tags can sometimes be meaningful only to the users that assigned them. They can be ambiguous and can also have a lot of synonyms. The authors of [14] developed a way of addressing these problems by expanding the tag set that is relevant to a user to include other related tags. They construct user profiles from the resulting weighted tags. Then, based on the same tag expansion procedure, they build item preference profiles (which include other related tags that are relevant for describing the item), under the assumption that tagged items are items that the user likes. Thus, they implicitly make item recommendations by combining a user based collaborative approach with a content based approach.

We expand the idea of weighted tags in the context of movie recommendations. We notice that tags may not always capture the true preference of the user (as assumed in [14]), therefore, in our scenario, we incorporate the actual ratings. Instead of simply counting the number of times user  $u_i$  has tagged an item with the tag  $t_x$ , we average the ratings  $u_i$  assigned to the movies tagged with  $t_x$  (by the same user  $u_i$ ). First, we define some notations. The user set  $U = \{u_1, u_2, \dots, u_{|U|}\}$  contains all the users that tagged movies in the hetrec2011-movielens-2k dataset. The movie set  $M = \{m_1, m_2, \dots, m_{|M|}\}$  contains all movies from the corpus, the tag set  $T = \{t_1, t_2, \dots, t_{|T|}\}$  contains all the tags used by the users in  $U$  to label movies in  $M$ . The approach can be organized into several components, as follows.

1) *Tag Relevance*: To lay the foundation for our WTRR, we first present how the *movie tag relevance weight* is calculated in [14]. Let  $m_i$  be a movie from  $M$ ,  $T_{m_i}$  is the set of all tags used by different users to describe the movie  $m_i$ .

For each tag  $t_x$  from  $T_{m_i}$ , the *movie tag relevance weight* is defined as  $w_{m_i}(t_x)$  and is calculated using the equation:

$$\text{WTR based: } w_{m_i}(t_x) = \frac{n_{m_i, t_x}}{\sum_{t_y \in T_{m_i}} n_{m_i, t_y}} \quad (1)$$

where  $n_{m_i, t_x}$  represents the number of times the tag  $t_x$  has been used by the users in the corpus to describe the movie  $m_i$ . The value of  $w_{m_i}(t_x)$  signifies how popular the tag  $t_x$  is for the movie  $m_i$ . This relevance metric reflects the *wisdom of crowds*. The higher the value of  $w_{m_i}(t_x)$ , the more likely it is that the tag  $t_x$  represents the topic of movie  $m_i$ .

We refine the metric in Equation (1) in our WTRR approach, by incorporating ratings rather than simple counts. To do this, we must ensure that the user who tagged the movies, also rated those movies. In other words, a movie must be both tagged and rated by a particular user.

$$\text{WTRR based: } w_{m_i}(t_x) = \frac{\sum_{u_j \in U_{m_i, t_x}} r_{u_j, t_x}(m_i)}{\sum_{u_j \in U_{m_i}, t_y \in T_{m_i}} r_{u_j, t_y}(m_i)} \quad (2)$$

where the numerator is a summation of the ratings assigned to the movie  $m_i$  by all the users who used  $t_x$  to annotate it. The set of users who used  $t_x$  to tag  $m_i$  is denoted by  $U_{m_i, t_x}$ . The denominator represents a summation of all the ratings from the users who tagged  $m_i$ . The value of  $w_{m_i}(t_x)$  now captures the true popularity of the tag  $t_x$  with respect to a movie  $m_i$ .

2) *Tag Relatedness Metric*: The relatedness metric between two tags is defined by  $c_{u_i}(t_x, t_y)$ , which represents the degree of correspondence (or connection) between  $t_x$  and  $t_y$  with respect to user  $u_i$ . It measures how similar tag  $t_y$  is to a given tag  $t_x$ , in the content of a user  $u_i$ . It is not a symmetric measure, in the sense that  $c_{u_i, t_x}(t_y)$  does not always equal  $c_{u_i, t_y}(t_x)$ . Tag relatedness metric is given by:

$$c_{u_i, t_x}(t_y) = \frac{1}{|M_{u_i, t_x}|} \sum_{m_j \in M_{u_i, t_x}} w_{m_j}(t_y) \quad (3)$$

where  $M_{u_i, t_x}$  is the set of movies tagged with  $t_x$  by  $u_i$ .

3) *User Profile*: In collaborative filtering, typically a user profile consists of items and their ratings. Usually, for comparing users in traditional collaborative filtering, direct user-to-user correlation is used and predictions are made by using historical rating data. In social networking, users tag those movies that they are most interested in (whether they like them or not). Explicit tag information given by users can be used to describe the users interests and preferences. In content-based approaches, users topic preferences are extracted from the content of items. To leverage the advantages of hybrid systems, users topic preferences and movie

preferences are combined. Thus, every user is represented by a profile, encoded using a vector of weights:

$$u_i = \{u_i^T, u_i^M\}$$

First,  $u_i^T$  is user  $u_i$ 's topic preferences and is represented by a  $|T|$ -sized tag vector with values denoting how much  $u_i$  is interested in each tag. These values are weights, for which we provide a formula in the following paragraphs (see Equation (8)). Secondly,  $u_i^M$  represents user  $u_i$ 's movie preferences and is represented by an  $|M|$ -sized movie vector, whose values are also weights (see Section III-B3).

We first introduce the definition of the vector  $u_i^T$  used in [14], where the assumption is that tags assigned by users explicitly describe the preferences of the users who assigned them. The number of times a tag is used by a user to describe a movie in the corpus shows how popular the tag is for that user. Therefore, it becomes necessary to capture the user tag preference (or user-tag relevance metric). The value of the metric signifies how strongly the user feels about a tag:

$$\text{WTR based: } w_{u_i}(t_x) = \frac{n_{u_i, t_x}}{\sum_{t_y \in T_{u_i}} n_{u_i, t_y}} \quad (4)$$

where  $T_{u_i}$  is the tag set of the user  $u_i$  and  $n_{u_i, t_x}$  is the number of movies that are collected under the tag  $t_x$  by user  $u_i$ , i.e., how many times the  $u_i$  has used tag  $t_x$ .

We hypothesize that  $u_i^T$  can be estimated more accurately from ratings. Since our dataset was reduced to a subset of movies for which all users provided both tags and ratings, we are able to change the above formula, as follows:

$$\text{WTRR based: } w_{u_i}(t_x) = \frac{\sum_{m_j \in M_{u_i, t_x}} r_{u_j, t_x}(m_j)}{\sum_{m_j \in M_{u_i, t_y \in T_{u_i}}} r_{u_i, t_y}(m_j)} \quad (5)$$

where the numerator is a summation of the ratings assigned to the movie  $m_j$  by all the users who used  $t_x$  to annotate it, and the denominator is the summation over all ratings assigned to the movie  $m_j$  by all the users who tagged it.

As tags related to  $t_y$  are believed to be representative for user  $u_i$ , the weight (relevance) of tag  $t_y$  for a user  $u_i$  is calculated as summation of relatedness between the tags used by user  $u_i$  (i.e.,  $t_x \in T_{u_i}$ ) and target tag  $t_y$ ;  $W_{u_i}(t_y)$  is the *total relevance weight* of  $t_y$  for  $u_i$  and is given by:

$$W_{u_i}(t_y) = \sum_{t_x \in T_{u_i}} w_{u_i}(t_x) \cdot c_{u_i, t_x}(t_y) \quad (6)$$

A tag's occurrence for all users must be taken into consideration in order to measure the *general* importance of a tag in the topic preference identification of a user;  $iuf(t_y)$  is the *inverse user frequency* of tag  $t_y$  is given by:

$$iuf(t_y) = \frac{1}{\log(e + |U_{t_y}|)} \quad (7)$$

$|U_{t_y}|$  is the number of users that used  $t_y$  and  $e$  is Euler's number, such that  $1 \geq iuf(t_y) \geq 0$ . Thus, the tag representation of each user is defined as below:

$$RU_T(u_i) = \{(w^{u_i}(t_y) \cdot iuf(t_y)) | t_y \in T\} \quad (8)$$

Equation (8) denotes the values of the topic preference vector  $u_i^T$  for each user  $u_i$ .

## B. WKRR

We now describe the keyword variant of our proposed approach. In this variant, our algorithm dynamically creates a user profile from IMDB movie keywords and explicit user ratings. IMDB allows users to provide keywords in a controlled manner, thus keyword descriptions of movies can be considered as consolidated "word of mouth". Our intuition is that such a profile captures better than tags a user's interests and preferences, as content features are more suitable for learning. The main goal here is to introduce a new algorithm for associating weights to keywords for each user individually, by using explicit user ratings.

Similar to WTRR, we profile each user  $u_i$  on user keyword topic preference and user rating-based movie preference, specifically  $u_i = \{u_i^K, u_i^R\}$ , where  $u_i^K$  is user  $u_i$ 's keyword topic preferences and is represented by a  $|K|$ -sized keyword vector with values denoting how much  $u_i$  is interested in a keyword;  $u_i^R$  is user  $u_i$ 's rating-based movie preferences and it is an  $|M|$ -sized rating vector that represents movie preference for each user. Further, we explain our approach for building user keyword topic preference that is derived from the user's ratings, as well as keywords from movies. We introduce some notations for better understanding of our approach. Let  $U = \{u_1, u_2, \dots, u_{|U|}\}$  be the set of all users,  $M = \{m_1, m_2, \dots, m_{|M|}\}$  be the set of all movies.  $K = \{k_1, k_2, \dots, k_{|K|}\}$  be the set of all the keywords used to annotate movies. Let  $R = \{0.5, 1, 1.5, 2, 2.5, 3, 3.5, 4, 4.5, 5\}$  be the set of all possible ratings users can give. Then we calculate the  $w_{u_i}(k_x)$  and  $iuf(k_x)$  with the constraint that for every weight measurement we only consider the keywords coming from a movie that has been rated with the same rating.

### 1) Movie Description Based on Weighted Keywords:

Similar to tags, keywords also face the problem of ambiguity. To overcome this problem we calculate  $w_{m_i}(k_y, r)$ , the movie keyword relevance metric, which is an estimate of the relevance of keyword  $k_y$  to the movie  $m_i$  in the context of rating  $r$ . In other words, how relevant is the keyword  $k_z$  with rating  $r$  to the movie  $m_i$ . For example, how relevant is "gun" as a preferred keyword to the movie "Matrix", as opposed to how relevant "gun" is as a disliked keyword to the movie "Matrix". The measure will capture the community's amenity/approval of a keyword toward a movie. Let  $m_i$  be the movie from  $M$  and  $K_{m_i}$  be the set of all the keywords used to represent movie  $m_i$ . Let  $n_{m_i, k_y, r}$  be the count of how many users rated the movie

$m_i$  which contains keyword  $k_y$  with rating  $r$ . For example, if  $k_x$  belongs to a movie that the user  $u_i$  rated with 4.5,  $n_{u_i, k_x}$  is the count of movies sharing keyword  $k_x$  which have been rated with either 4, 4.5 or 5 by any of the other users. The movie keyword relevance metric  $w_{m_i}(k_y, r)$  is given by:

$$w_{m_i}(k_x, r) = \frac{n_{m_i, k_x, r}}{\sum_{k_y \in K_{m_i}} n_{m_i, k_y, r}} \quad (9)$$

2) *The Representation of Keywords*: Keywords also suffer from the problem of synonymy, so it is necessary to find the semantic meaning of each keyword for all possible ratings for every users individually. The intuition behind this approach is that if a user has rated a movie  $m_i$  with a high value, that means that keywords associated to  $m_i$  are important to that user. To evaluate the importance of keyword  $k_x$  for user  $u_i$  who rated  $m_i$  with  $r$ , we aggregate the movies containing  $k_x$  and having rating  $r$  together, to form topic preference. Since the movie topic can be described by weighted keywords, keywords from all the movies containing  $k_x$  and rated with  $r$  can be used to describe the topic preference of user  $u_i$ . By this, we capture related keywords to calculate the topic preferences. We define  $c_{u_i, k_x}(k_y, r)$  to represent the degree of correspondence (or connectivity) between keywords  $k_x$  and  $k_y$  with respect to user  $u_i$  in the context of rating  $r$ ;  $|M_{u_i, k_x, r}|$  is the number of movies containing keyword  $k_x$  that user  $u_i$  has rated with rating  $r$ :

$$c_{u_i, k_x}(k_y, r) = \sum_{m_j \in M_{u_i, k_x}} \frac{1}{|M_{u_i, k_x, r}|} \cdot w_{m_j}(k_y, r) \quad (10)$$

The representation of keyword  $k_x$  for user  $u_i$  consists of a set of keywords related to  $k_x$  along with their corresponding weights for rating  $r$ :

$$RK(u_i, k_x, r) = \{(k_y, c_{u_i, k_x}(k_y, r)) | k_y \in K\} \quad (11)$$

where  $\sum_{k_y \in K} c_{u_i, k_x}(k_y, r) = 1$ .

We now can build the user profile which is free of tag ambiguities and tag synonymy.

3) *User Profile Generation From Keywords*: In this section we propose a new method to calculate the user keyword topic preference. Our goal is to estimate the importance of a keyword to a user, given a particular rating. First, we calculate the keyword relevance metric which shows how strong a keyword  $k_x$  is relevant to a user when the movies described by  $k_x$  are rated with  $r$ .

$$w_{u_i}(k_x, r) = \frac{n_{u_i, k_x, r}}{\sum_{k_y \in K_{u_i}} n_{u_i, k_y, r}} \quad (12)$$

where  $n_{u_i, k_x, r}$  is the number of movies that share the keyword  $k_x$  and are rated by user  $u_i$  with a rating value  $r$ , in other words, how many times the user has rated a movie

containing keyword  $k_x$  with a rating value equal to  $r$ .  $K_{u_i}$  is the keyword set of the user  $u_i$ . The total relevance weight of a keyword for a user  $u_i$  is given by:

$$W_{u_i}(k_x) = \sum_{k_x \in K_{u_i}, r \in R} w_{u_i}(k_x) \cdot c_{u_i, k_x}(k_y) \cdot iuf(k_y) \quad (13)$$

The *inverse user frequency*  $iuf(k_x, r)$  of keyword  $k_x$  rated with a rating  $r$ : A keyword's occurrence within movies rated by all users must be taken under consideration in order to measure the *general* importance of a keyword in the topic preference of a user. This is how we calculate the inverse user frequency:

$$iuf(k_x, r) = \frac{1}{\log(e + |U_{k_x, r}|)} \quad (14)$$

where  $|U_{k_x, r}|$  is the number of users that rated movies which contain  $k_x$  with a rating value in  $r$  and  $e$  Euler's number.

### C. Neighborhood Formation

In order to predict how much a user will enjoy an unseen movie, in other words to predict their rating for it, we first set out to find the community of users sharing similar taste, a.k.a.  $k$  nearest neighbors. The main goal is to identify for each user  $u$ , an ordered list of  $k$  most similar users,  $U = \{u_1, u_2, \dots, u_k\}$  such that  $u \in U$  and  $sim(u, u_1)$  is maximum,  $sim(u, u_2)$  is the second highest and so on.  $K$ -nearest users are selected based on the similarity value. The similarity values play a double role in neighborhood-based recommendation methods:

- they allow the selection of trusted neighbors whose ratings are used in the prediction, and
- they provide the means to give more or less importance to these neighbors in the prediction

The selection of the similarity measure is one of the most critical aspects of building a neighborhood-based recommender system, as it can have a significant impact on both its accuracy and its performance. Thus, for computing the similarity between all users or the similarity between movies, the cosine similarity measure is used. In this case, two users are thought of as two vectors in the  $T$ -dimensional user space. The similarity between them is measured by computing the cosine of the angle between the two vectors representing them.

As discussed in Section III-A3, each user is encoded with their own topic preferences and movie preferences. The similarity between two users based on user topic preference is denoted as  $sim_u^T(u_i, u_j)$ , whereas the similarity between two users based on user movie preference is denoted as  $sim_u^M(u_i, u_j)$  where  $T$  and  $M$  are the sets of all tags and movies, respectively.

The similarity between two users is given by:

$$sim(u_i, u_j) = \omega \cdot sim_u^T(u_i, u_j) + (1 - \omega) \cdot sim_u^M(u_i, u_j) \quad (15)$$

where  $0 \leq \omega \leq 1$ .

The parameter  $\omega$  controls the extent of the collaborative dimension of the algorithm. As we decrease the value of  $\omega$ , the algorithm will be predominantly collaborative, as the contribution of the users movie preferences will dominate. We believe our algorithm can also be helpful in alleviating the problem of cold start. During the experimental phase, we kept  $\omega = 0.9$ .

Since we have the similarities between different users, a set of similar neighbors can be identified. The traditional Top N algorithms choose the Top N most similar neighbors to predict the missing value, which in our case is a prediction for a movie which is not yet watched by a user. To predict a missing value  $r(u,m)$  in the movie-user matrix, a set of users similar to  $u$ , specifically  $N(u)$ , is denoted by:

$$N(u) = \{v | v \in T(u), u \in U\} \quad (16)$$

where  $T(u)$  is the set of N most similar users to user  $u$ .

#### D. Rating Prediction Formula

To calculate the missing ratings we used a popular user-based prediction formula described in [11]. The intuition behind this prediction scheme is that user rating distributions spread around different points. E.g., one user rates a good movie with 4 and a bad movie with 2, whereas others users are using 1 for bad movies and 3 for good movies. Intuitively, different users judge movies differently, thus user ratings are infrequent. This prediction scheme normalizes the rating  $r(u,m)$  by dividing the user-mean-centered rating by the standard deviation  $\sigma_u$  of the ratings given by user  $u$ .

$$\hat{r}(u,m) = \bar{r}_u + \sigma_u \frac{\sum_{v \in N(u)} \frac{w_{uv}(r_{v,m} - \bar{r}_v)}{\sigma_v}}{\sum_{v \in N(u)} |w_{uv}|} \quad (17)$$

Following the notation from [11],  $\bar{r}_u$  is the average of the ratings given by user  $u$ ,  $w_{uv}$  is the similarity value between user  $u$  and user  $v$ ,  $\sigma_u$  is the standard deviation of ratings given by user  $u$  and finally,  $N(u)$  is set of most similar users to user  $u$ .

## IV. EXPERIMENTAL SETUP

In this section, we provide more information on the dataset used in Section IV-A, we describe the metrics used for evaluation in Section IV-B, and explain how we conducted our experiments in Section IV-C.

### A. DataSet

The data set used in our experiments, hetrec2011-movieleens-2k dated May 2011, is made available to the public by [7]. It is based on the original MovieLens10M dataset, published by the GroupLens<sup>2</sup> research group. The movies in this data set are also referencing their corresponding web pages at the IMDB website. More information

<sup>2</sup><http://www.grouplens.org>

Table I  
STATISTICS ABOUT HETREC2011-MOVIIELENS-2K DATASET.

# users	2,113
# movies	10,197
# movie genres	20
# actors	95,321
# countries	72
# tags	13,222
# tag assignment	47,899
# ratings	855,598

about the format as well as statistics regarding the data are available at the hetrec2011-movieleens-2k website. To summarize, there are 2,113 users, 10,197 movies and a total of 13,222 unique tags that fall into 47,957 tag assignment tuples of the form [user, tag, movie]. There are also 855,598 user ratings ranging from 0.5 to 5.0, in increments of 0.5, thus a total of 10 distinct rating values. There is an average of 405 ratings per user, and 85 per movie. The density of the hetrec2011-movieleens-2k movie dataset is 3.97%. This data set has been previously used in [6], [18] and [12]. In addition to the hetrec2011-movieleens-2k dataset, for our WKRR variant we also used content information about the movies from IMDB.

Table I shows some statistics about the hetrec2011-movieleens-2k dataset.

Our experimental setup requires some pre-processing of the data. For comparing WTRR and WKRR, we need the movies that users have both tagged and rated. After taking the intersection between movies that are rated and movies that are tagged by users, we obtained a subset of 4,655 movies for which every user provided a tag as well as a rating. This subset contains 762,238 tag/rating assignments from 1,097 users. A number of 8,288 unique tags were used to make 762,238 tag assignments. The density of this dataset is 14.93%.

### B. Evaluation Metrics

Evaluating the recommender system and determining the accuracy of an algorithm is difficult for several reasons. First, algorithms behave differently on different datasets. Second, the goals of the recommenders may vary and evaluation will be based on fulfilling unique criteria. Thus, according to [11], accuracy measures for recommendation systems can be classified into the following categories: predictive accuracy metrics, such as Root Mean Squared Error (RMSE) or Mean Absolute Error (MAE) and its variations; classification accuracy metrics, such as precision, recall, F1-measure and ROC sensitivity; rank accuracy metrics, such as Mean Average Precision (MAP), normalized distance-based performance metric (NDPM), etc.

To measure the accuracy of our algorithms, we use the RMSE measure. This also allows us to compare our results

with various others. The RMSE value is given by:

$$RMSE = \sqrt{\frac{1}{N} \sum_{u,m} (p_{u,m} - r_{u,m})^2} \quad (18)$$

where  $N$  is the total number of ratings from all users,  $p_{u,m}$  is the predicted rating for user  $u$  on movie  $m$ , and  $r_{u,m}$  is the actual rating for movie  $m$  assigned by the user  $u$ . RMSE amplifies the contributions of the absolute errors between the predictions and the true rating values.

### C. Experiments

During the experimental phase of our study, to avoid bias, we split the original rating data into training and test sets. We used 5-fold cross validation to avoid any fortunate occurrences. We trained our algorithm on the train set and then predicted the ratings in the test set. Thus, for each of the 5 splits of our hetrec2011-movieleens-2k dataset, we kept 80% of users for training, while 20% of users were set aside for test. For each user in the test set, we then hid 20% of the movie ratings while the remaining 80% of movie ratings were used for training.

- First, we ran experiments to compare WTRR and WKRR with the purely collaborative technique, on a subset of the dataset, which contains movies that both tagged and rated.
- Second, we assessed the applicability of WKRR on the whole dataset, in order to compare to other results available in the recent literature.

## V. RESULTS

We now present and discuss the performance of our proposed approach. The experiments were designed to investigate the effectiveness of using ratings along with tags or keywords, for the hetrec2011-movieleens-2k. Given that keyword features allow us to use a larger dataset as compared to tag features, we hypothesize that user profiles based on keyword result in better accuracy overall, when all data is used. However, given that tags are more specific to a user, we hypothesize that the tag based approach performs better on a dataset that has both tags and keywords. Note that in this study we used normalized user-mean-centered rating by standard deviation for prediction calculation.

We first compare the two variants, WTRR and WKRR, of the proposed approach. To be able to do this comparison, we ran tests on the subset of the data that contained both ratings and keywords. As a baseline, we used the purely collaborative (PC) approach, where we used the ratings to create the movie profile. To study the impact of the neighborhood size, we also ran the experiments with  $k \in \{10, 20, 30, 50, 75\}$  for both WTRR and WKRR with the actual ratings, and compared the results in terms of accuracy (RMSE). The values are displayed in Table II. The prediction quality improves with the increase in the neighborhood size in all cases. Both variants of our approach

Table II  
COMPARISON BETWEEN WTRR, WKRR AND A CF BASELINE: RMSE VALUES BASED ON EXPERIMENTS RUN WITH AN INCREASING NUMBER (N) OF USERS IN THE NEIGHBORHOOD FORMATION.

N	PC	WTRR	WKRR
75	0.82948	0.81123	0.80727
50	0.83829	0.82856	0.81883
30	0.85460	0.84770	0.83986
20	0.84331	0.83145	0.83809
10	0.86983	0.85731	0.84698

Table III  
PERFORMANCES OF THE WKRR VARIANT OF OUR PROPOSED APPROACH, BY COMPARISON WITH OTHER RESULTS FROM RECENT WORK ON THE COMPLETE HETREC2011-MOVIIELEENS-2K DATASET. PERFORMANCES FOR IMBRF, PURE CF, CA, LLS AND AVGR WERE OBTAINED FROM [6], WHILE THE PMF VALUE IS FROM [12].

WKRR	IMBrf	CF	CA	AVGR	LLS	PMF
0.8304	0.8797	0.8876	0.9436	1.088	0.8758	0.8367

outperform the purely collaborative baseline. Furthermore, the WTRR variant shows improvements over WKRR.

For many real world datasets, a lot of movies don't have user tags, which make impractical tag-based approaches. To be able to use that information, instead of using the exact tag, we use keywords, which represent a collective categorization of movies. This allows us to use the hetrec2011-movieleens-2k dataset in its entirety. In Table III, we compare the results of the WKRR variant of our approach with the results of state of the art approaches reported in [6] and [12]. The results of this comparison reinforce our intuition that assigning different weights to keywords based on rating information is useful, as the WKRR results are better than all results previously reported in the literature. The PMF approach is very close in performance to our approach. While that approach is more sophisticated and computationally more expensive, our approach makes use of extra information in form of keywords extracted from IMDB [12].

## VI. CONCLUSIONS

In this work we propose a novel hybrid recommendation technique for combining the collaborative filtering and the content based recommendation techniques, and show that it outperforms other approaches in the neighborhood category. Our approach has two variants: WTRR and WKRR, which use tags and keywords, respectively, to capture content, and explicit ratings to capture collaborative filtering. Our proposed hybrid recommender system was built using the hetrec2011-movieleens-2k dataset, supplemented with extra movie information from the IMDB online archive in the case of the WKRR variant. We alleviate the noise and synonymy problems of keywords by considering a pool of related terms when constructing the profiles. Secondly, to use the collaborative based filtering, user-user similarities

are calculated. Using the similarity estimation approach,  $k$ -nearest neighbors are found and based on their relatedness, and we determine ratings for unseen movies. The results of our experiments show that the performance of WKRR exceeds the other approaches, when the whole dataset is used. However, WTRR is better than WKRR, when only the subset of data with both tags and keywords is used. Our results show that for datasets in which relevant tag information is scarce, extending the features from tags to movie keywords and ratings boosts the performance.

This type of approach is applicable to any domain that has item descriptions and users willing to rate the items. In future work, we plan on using larger versions of the MovieLens<sup>3</sup> datasets, for example MovieLens1M, consisting of 1 million ratings from 6000 users on 4000 movies, or even MovieLens10M which comprises 10 million ratings and 100,000 tag applications applied to 10,000 movies by 72,000 users. For this approach, the Map-Reduce framework might be a good environment or reducing the dimension of matrices by means of singular value decomposition. This work focuses on profiling each user based on ratings but we can also profile movies in same feature space so that similarity between user and movies is used for making recommendations. Applying an improved customized prediction scheme would also be a good addition. Using our approach on clusters of users with similar taste may also improve the relevance of the predictions.

#### REFERENCES

- [1] 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*, 17(6):734–749, 2005.
- [2] X. Amatriain, J. Pujol, and N. Oliver. I like it... i like it not: Evaluating user ratings noise in recommender systems. In *User Modeling, Adaptation, and Personalization*, Lecture Notes in Computer Science. 2009.
- [3] M. Balabanovic and Y. Shoham. Fab: content-based, collaborative recommendation. *Communications of the ACM*, 40, 1997.
- [4] L. Balby Marinho, A. Hotho, R. Jäschke, A. Nanopoulos, S. Rendle, L. Schmidt-Thieme, G. Stumme, and P. Symeonidis. Recommender systems. In *Recommender Systems for Social Tagging Systems*, SpringerBriefs in Electrical and Computer Engineering. Springer US, 2012.
- [5] R. M. Bell, Y. Koren, and C. Volinsky. The Bellkor 2008 solution to the Netflix prize. 2008.
- [6] E. Bothos, K. Christidis, D. Apostolou, and G. Mentzas. Information market based recommender systems fusion. In *Proceedings of the 2nd International Workshop on Information Heterogeneity and Fusion in Recommender Systems*, 2011.
- [7] I. Cantador, P. Brusilovsky, and T. Kuflik. 2nd workshop on information heterogeneity and fusion in recommender systems (hetrec 2011). In *Proceedings of the 5th ACM conference on Recommender systems*, 2011.
- [8] M. Claypool, A. Gokhale, T. Miranda, P. Murnikov, D. Netes, and M. Sartin. Combining content-based and collaborative filters in an online newspaper. In *In Proceedings of ACM SIGIR Workshop on Recommender Systems*, 1999.
- [9] M. Clements, A.P. De Vries, and M.J.T. Reinders. The task-dependent effect of tags and ratings on social media access. *ACM Trans. Inf. Syst.*, 28(4):21:1–21:42, November 2010.
- [10] M. de Gemmis, P. Lops, G. Semeraro, and P. Basile. Integrating tags in a semantic content-based recommender. In *Proceedings of the 2008 ACM conference on Recommender systems*, RecSys '08. ACM, 2008.
- [11] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl. Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems*, 2004.
- [12] C. Jones, J. Ghosh, and A. Sharma. Learning multiple models for exploiting predictive heterogeneity in recommender systems. 2011.
- [13] V. Kostakos. Is the crowds wisdom biased? a quantitative analysis of three online communities. IEEE Computer Society, 2009.
- [14] H. Liang, Y. Xu, Y. Li, R. Nayak, and G. Shaw. A hybrid recommender systems based on weighted tags. *10th SIAM International Conference on Data Mining*, 2010.
- [15] S. Nakajima and M. Sugiyama. Implicit regularization in variational bayesian matrix factorization, 2007.
- [16] D. H. Park, H. K. Kim, Il. Y. Choi and J. K. Kim. A literature review and classification of recommender systems research In *Expert Systems with Applications*, volume 39, 2012
- [17] P. Resnick and H. R. Varian. Recommender systems. *Communications of the ACM*, 1997.
- [18] A. Said, B. Kille, E. W. De Luca, and S. Albayrak. Personalizing tags: a folksonomy-like approach for recommending movies. In *Proceedings of the 2nd International Workshop on Information Heterogeneity and Fusion in Recommender Systems*, HetRec '11, 2011.
- [19] J. B. Schafer, J. Konstan, and J. Riedl. Recommender systems in e-commerce. In *Proceedings of the 1st ACM conference on Electronic commerce*. ACM, 1999.
- [20] X. Su and T. M. Khoshgoftaar. A survey of collaborative filtering techniques. *Advances in Artificial Intelligence*, 2009.
- [21] J. Surowiecki and M. P. Silverman. The wisdom of crowds. *American Journal of Physics*, 2007.

<sup>3</sup><http://www.grouplens.org/node/12>