Predicting the Ratings of Multimedia Items for Making Personalized Recommendations

Rani Qumsiyeh
Computer Science Department
Brigham Young University
Provo, Utah 84602
rani_qumsiyeh@yahoo.com

Yiu-Kai Ng
Computer Science Department
Brigham Young University
Provo, Utah 84602
ng@compsci.byu.edu

ABSTRACT

Existing multimedia recommenders suggest a specific type of multimedia items rather than items of different types personalized for a user based on his/her preference. Assume that a user is interested in a particular family movie, it is appealing if a multimedia recommendation system can suggest other movies, music, books, and paintings closely related to the movie. We propose a comprehensive, personalized multimedia recommendation system, denoted MudRecS, which makes recommendations on movies, music, books, and paintings similar in content to other movies, music, books, and/or paintings that a MudRecS user is interested in. MudRecS does not rely on users' access patterns/histories, connection information extracted from social networking sites, collaborated filtering methods, or user personal attributes (such as gender and age) to perform the recommendation task. It simply considers the users' ratings, genres, role players (authors or artists), and reviews of different multimedia items, which are abundant and easy to find on the Web. MudRecS predicts the *ratings* of multimedia items that match the interests of a user to make recommendations. The performance of MudRecS has been compared with current state-of-the-art multimedia recommenders using various multimedia datasets, and the experimental results show that MudRecS significantly outperforms other systems in accurately predicting the ratings of multimedia items to be recommended.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Information filtering

Keywords

Multimedia recommender, rating, genre, review, popularity

1. INTRODUCTION

Current multimedia recommenders either integrate user group information extracted from social websites, consider user profiles or attributes (such as gender and age), or analyze past behavior of individual users or overall behavior of a community of users to make recommendations. These information sources are not always available to be downloaded using APIs over the Internet or social networking sites. Moreover, majority of the existing multimedia recommenders suggest only a particular type of multimedia items, which are either videos, audios, text, or images. The restricted type of recommended items is a deficient and limitation, since various types of multimedia items offer different sources of information that are appealing to extended groups of users. In this paper, we propose a personalized multimedia recommendation system, denoted MudRecS, which recommends four different types of multimedia items—movies (videos), music (audios), books (text), and paintings (images)—for its users based on their personal preferences.

MudRecS relies on neither training data, predefined knowledge bases, nor ontologies to make recommendations. It does not require any manual intervention from its users nor users' personal information. Instead, it devises recommendations solely based on users' ratings, genres, reviews, popularity of artists/authors, and readability measures on multimedia items Is that are vastly available on the Web. Besides considering the *genres* of Is, MudRecS extracts *features* from *reviews* and determines the *polarity* and general *opinions* on the features of Is. MudRecS presents a unique approach for configuring the *popularity* of role players (authors/artists) of Is that is utilized for devising the *user preference* on Is. Moreover, whenever applicable, MudRecS analyzes the *comprehension levels* of its users so that it enhances its predictions on Is that are appealing to the users.

The design of MudRecS, which compares favorably to current state-of-the-art collaborative filtering, content-based, or hybrid approaches for generating recommendations, is simple and easy to develop. It is a contribution to the area of study in multimedia recommendation, since it introduces novel measures that accurately *predict* the *ratings* of multimedia items of various domains for each individual user. Predicted ratings of items can be used for ranking the recommended items, and items with higher predicted ratings are expected to be more favorable to the users who have previously rated items of the same or a different domain, which constitute the basis for rating prediction.

We organize our paper as follows. In Section 2, we discuss previous work on multimedia recommenders. In Section 3, we detail the design of MudRecS. In Section 4, we present the experimental results which were used to evaluate the performance of MudRecS. In Section 5, we give a conclusion.

SIGIR'12, August 12–16, 2012, Portland, Oregon, USA.

2. RELATED WORK

Machine learning, information retrieval, natural language processing, and probabilistic techniques have been adapted in the past to develop systems that recommend movies [9], song/music tracks [3], text documents [7], and images [13], to name a few. Majority of existing recommendation systems are either content-based or collaborative-filtering-based. The content-based (or cognitive) approaches [2] create a profile to capture the (items of) interest of a user U using words, phrases, and/or features. Recommenders that are content-based identify the common features of items with favorable ratings given by a user and recommend to the user other items that share the same or similar features. Recommendation systems purely based on content similarity measure, however, generally suffer from the problems of limited-content analysis and over-specialization [4]. Limited-content analysis is a result of the relatively small amount of information on users or the content of items, whereas over-specialization restricts the number and diversity of items to be recommended, since the predicted rating of an item I for a user U is *high* only if I *matches* in content (such as genres or features) with the ones that U likes.

Collaborative filtering is another well-known recommendation method [14] which identifies the group of people who share common interest or similar items with user U and recommends items to U based on the group's interests. Amazon's recommender, which applied the collaborative-filtering technique [11], suggested items to a user based on other users' previous purchase patterns and/or rated items. Such a system is differed from MudRecS, since the latter does not consider users' access patterns nor histories.

Various hybrid approaches [2] have been introduced which exploit the benefits of using both collaborative-filtering and content-based methods to improve the quality of recommendations. Contractually, MudRecS considers semantic information, such as genres, reviews, popularity, and comprehensibility of multimedia items to make recommendations.

A method that translates tags assigned to multimedia contents for cross-language retrieval is proposed in [13]. The lexical translation of annotated tags, which handles sense disambiguation of tags, however, is restricted to image retrieval only and relies on social tagging, which is not required by MudRecS. An agent-based multimedia recommendation system, which is also constrained to process image data, is presented in [10]. The recommender depends on a predefined ontology to capture the users' preferences and identifies the semantics of hypermedia images to derive their contextual information for making recommendations. MudRecS avoids constructing ontologies and accessing hyperlinks to assess multimedia items to be recommended, which eliminates the tedious setup and avoids imposed overheads.

3. OUR RECOMMENDATION SYSTEM

Multimedia items on the Web are rated using the standard "star rating system" in which an item is given a number of stars (a rating level) by web users. To predict the ratings of items previously unrated by a user U for making recommendations to U, MudRecS considers different sources of information—U's ratings, genres, reviews, role players, and text readability levels—on items, which are either movies, songs, books, and/or paintings, previously rated by U in two steps: analysis of pre-rated items and prediction.

Analysis. Given a set of multimedia items previously rated by a user U, MudRecS analyzes the preferences of U based on the genres, features, and role players of the items that U (dis)likes. The *genres* and *role players* of multimedia items can easily be extracted from representative and comprehensive multimedia websites, such as imdb.com for movies, last.fm for songs, iblist.com for books, and flickr.com for images, using a simple HTML parser, whereas *features* of multimedia items can be retrieved from reviews that are available at well-known websites, such as Epinions(.com), Consumersearch(.com), and Consumerreports(.org). All of the rating information of U are archived in MudRecS for future references. (See tables in Figures 1, 5, and 6 and Table 2 for sample archived data.) Since the number of multimedia items rated by U is only in the hundreds, or thousands the most, this analysis step is not a time-consuming nor labor-intensive process. Moreover, analysis on U is conducted once after which ratings can be predicted for items unrated by U.

Prediction. Using the set of analyzed multimedia items S rated by U, MudRecS predicts the rating of an item I unrated by U. MudRecS considers the *genres* applied to I, using previous *ratings* given by U on different genres to determine the *genre score*, denoted GS (discussed in Section 3.1), of I. MudRecS also considers each *feature* of I extracted from the reviews of I and computes the *review score*, denoted RwS, for I using *reviews* on items in S (detailed in Section 3.2). According to the *popularity* of each role player, which is either the author, actor, actress, director, or artist, of I, MudRecS assigns a *role player score*, denoted RPS, to I (see Section 3.3). Last, but not least, MudRecS calculates the *readability score*, denoted ReS, for I, if I is a book (presented in Section 3.4). MudRecS combines GS, RwS, RPS, and ReS of I to compute the $Rating\_Score$ of I using the *Stanford Certainty Factor* (SCF) [12] at each one of the N rating levels R_i ($1 \leq i \leq N$), where N is often in the range of 5 to 10 (as defined in Equation 1). The *rating level* with the *highest Rating_Score* is selected as the predicted rating for I, which can be used to determine the *ranking* order of recommended items unrated by U.

$$MAX_{i=1}^{N}\{Rating\_Score(R_i, I) = (GS(R_i, I) + RwS(R_i, I) + RPS(R_i, I) + ReS(R_i, I, L)) \;/\; (1 - Min\{GS(R_i, I), RwS(R_i, I), RPS(R_i, I), ReS(R_i, I, L)\})\} \quad (1)$$

where L is the readability level of I. If I is not a text document, then $ReS(R_i, I, L)$ is excluded from Equation 1 when $Rating\_Score(R_i, I)$ is computed.

3.1 The Genres of Multimedia Items

A genre is the *category* of an item I. (Table 1 shows sample genres for movies, music, books, and paintings, respectively.) The basic assumption of genres is that if two items have the *same* genre, then they are likely given *similar* ratings by the same user [6]. Besides determining the genres of I, MudRecS also considers *genre similarity* for rating prediction on I, since (i) two different genres can be highly similar, such as "suspense" and "horror" in movies, "hip hop" and "rap" in music, "dark" and "mystery" in books, and "pointillism" and "divisionism" in images, (ii) same items are sometimes assigned different genres by different experts/users, and (iii) some items have multiple genres assigned to them. *Similarities* of various genres can be determined using word-

Multimedia Items	Sample Genres
Movies	Action, Adventure, Comedy, Crime, Drama
Music	Classical, Heavy Metal, Jazz, Rap, Rock
Books	Crime, Fantasy, Fiction, Mystery, Romance
Paintings	Art, History, Landscapes, Life, Portraits

Table 1: Sample genres for different types of multimedia items considered by MudRecS

correlation factors (defined in Section 3.1.1). For each rating level of an item I, MudRecS computes the $Genre\_Score$ (GS) for each possible genre of I (as detailed in Section 3.1.2).

3.1.1 Similarities Among Different Genres

To determine the similarity of two genres, we employ the word-correlation factors in the *word-similarity* matrix [8], denoted WS-*matrix*. The similarity, denoted $Word\_Sim$, of any two non-stop, stemmed words i and j in WS-*matrix* is computed using the (i) *frequency* of co-occurrence and (ii) relative *distances* of i and j in each document in which they occur. WS-*matrix* was constructed using the documents in the Wikipedia collection (en.wikipedia.org/wiki/Wikipedia: Databasedownload) with 930,000 documents written by more than 89,000 authors on various topics and writing styles.

Compared with WordNet (wordnet.princeton.edu) in which each pair is not assigned a similarity weight, word-correlation factors offer a more sophisticated measure of word similarity.

3.1.2 Computing Genre Scores

Using the set of multimedia items S of the same domain D previously rated by a user U and an item I of D, which is unrated by U with various genres, MudRecS assigns GS of I at the rating level R_i ($1 \leq i \leq N$) as

$$GS(R_i, I) = \frac{\sum_{G \in G_{Set}} GSG(R_i, G)}{|G_{Set}|}, \text{ where}$$

$$GSG(R_i, G) = \frac{|Items\ in\ S\ with\ G\ of\ I\ Rated\ at\ R_i\ by\ U|}{|Items\ in\ S\ Rated\ at\ R_i\ by\ U|} + \sum_{j=1, G_j \neq G}^{|G_{Set}|} \frac{|Items\ in\ S\ with\ G_j\ of\ I\ Rated\ at\ R_i\ by\ U|}{|Items\ in\ S\ Rated\ at\ R_i\ by\ U|} \times Word\_Sim(G, G_j) \quad (2)$$

where G_{Set} is the set of genres of I, $G_j \in G_{Set}$ ($1 \leq j \leq |G_{Set}|$), and $Word\_Sim(G, G_j)$ is the genre similarity of G and G_j. Note that $Word\_Sim(G, G_j) = Word\_Sim(G_j, G)$.

Equation 2 assigns a higher GSG to a genre G at R_i if G in S has been rated at R_i by U more frequently than at any other rating levels in S. The multiplication is used to assign weight to genres such that the *less similar* a genre G_j is to G, the *less* it is *weighted* in the computation of $GSG(R_i, G)$. Figure 1 shows an example of applying Equation 2 to a number of genres in movies, which illustrates how the 'horror' genre receives a higher score at rating level 1 than at rating level 5, since most 'horror' movies, along with genres highly similar to 'horror', i.e., 'thriller', were rated at level 1. On the other hand, 'action' and its highly similar genre, 'Adventure,' were rated higher at level 5 than at level 1.

We realize that solely relying on the genres of an item to make recommendations could yield poor rating predictions, since (i) a *large* amount of data are required to accurately determine a particular user's genre preferences, (ii) a poor

Rating	Horror (*H*)	Thriller (*T*)	Action (*Ac*)	Adventure (*Ad*)
1	50	40	6	4
2	20	9	8	4
3	4	7	7	7
4	3	8	15	5
5	1	3	25	18

$WS(H, T) = 0.93$
$WS(H, Ac) = 0.35$
$WS(H, Ad) = 0.15$
$WS(T, Ac) = 0.67$
$WS(T, Ad) = 0.33$
$WS(Ac, Ad) = 0.91$

$$GSG(1, H) = \frac{50}{100} + \left(\frac{40}{100} \times WS(H,T) + \frac{6}{100} \times WS(H,Ac) + \frac{4}{100} \times WS(H,Ad)\right) = \mathbf{0.90}$$

$$GSG(5, H) = \frac{1}{47} + \left(\frac{3}{47} \times WS(H,T) + \frac{25}{47} \times WS(H,Ac) + \frac{18}{47} \times WS(H,Ad)\right) = 0.32$$

$$GSG(1, Ac) = \frac{6}{100} + \left(\frac{50}{100} \times WS(Ac,H) + \frac{40}{100} \times WS(Ac,T) + \frac{4}{100} \times WS(Ac,Ad)\right) = 0.54$$

$$GSG(5, Ac) = \frac{25}{47} + \left(\frac{1}{47} \times WS(Ac,H) + \frac{3}{47} \times WS(Ac,T) + \frac{18}{47} \times WS(Ac,Ad)\right) = \mathbf{0.93}$$

Figure 1: Four GSG scores computed at two rating levels for an unrated movie based on movies previously rated by a user U, and WS stands for $Word\_Sim$

(good, respectively) rating value for an item I with genre G does not necessarily indicate that a user dislikes (likes, respectively) G, but may have given a poor (good, respectively) rating to I due to other factors, (iii) items can be highly similar with respect to their genres and being dissimilar with respect to other features not related to genres, such as the 'story' of a movie. MudRecS considers the genres of I as well as other features addressed in online reviews on I.

3.2 Reviews

In predicting the rating of an item I, consumer reviews, opinions, and shared experiences on I are valuable sources of information which reflect consumers' preferences on I. Unlike user ratings that simply indicate their degrees of appealing on items, *reviews* provide more detailed information. Positive/Negative opinions on different *features* (i.e., properties) of I are often included in a review of I, such as the (originality of the) story in a movie, the (innovative or mono) tone of a song, or the (surprise or predictable) ending of a book. This information can be used to enhance the accuracy of rating prediction in a recommendation system.

To utilize the valuable information included in the set of reviews RS on items previously rated by user U for predicting the rating level of I, MudRecS (i) retrieves a set of reviews on I from review search engines (as detailed in Section 3.2.1), (ii) relies on a feature-detection approach to identify features of I from retrieved reviews, and (iii) determines the *polarity* of each sentence in reviews, which has been clustered based on a particular feature, as either positive or negative (in Section 3.2.2). Hereafter, each identified feature F of I is assigned the *Polarity Feature Score* (PFS) at each rating level based on the number of positive/negative sentences in RS assigned to F at certain rating levels (in Section 3.2.3).

3.2.1 Retrieving Relevant Reviews

MudRecS retrieves reviews for I from Epinions, Consumersearch, and Consumerreports. Using a query Q with keywords in the item name of I, MudRecS selects the top-33 reviews extracted from each one of the three review repositories[1] retrieved in response to Q, which yields the set of reviews of I for identifying the (polarity of) features of I.

[1]MudRecS retrieves 33 reviews from each repository, if they exist, since a collection of a hundred reviews on I is an ideal set for analyzing the features of I [5].

The film is shot using mainly unknown **actors**, who are very good in their parts. The film works as a totally **visual experience**, which I thought was one of the best I have seen in a while. The well-chosen words of the **script** are amazing and are almost incapable of adding to the unspoken pageantry and majesty of the remarkable **cinematography**. There is really nothing much about the **plot** of the movie; you either know it or you don't, but without new or added things. The **directing** of the film was good as well as original. The director was able to mix the spiritual battle that rages between Christ and Satan, while keeping the viewer informed of the **story of the movie** in reality.

Figure 2: An Epinions.com review for the movie "Passions of the Christ" and its identified features

Visual Graphics. The film works as a totally visual experience, which I thought was one of the best I have seen in a while. Strong acting and impressive visuals make The Passion of The Christ one of the best ...

Script. The well-chosen words of the script are amazing and are almost incapable of adding to the unspoken pageantry and majesty of the remarkable cinematography. As for the script of Mel Gibson's drama, ...

Director. The directing of the film was good as well as original. ...

Story. The director was able to mix the spiritual battle that rages between Christ and Satan, while keeping the viewer informed of the story ...

Figure 3: Some features/sentences extracted from the reviews of the movie "Passions of the Christ"

3.2.2 Detecting Item Features and Opinions in Reviews

Since each review R in the set of 99 retrieved reviews on an item I may address various features of I, MudRecS identifies distinct *features* of I and *clusters* sentences in R based on the detected features. Sentences in a feature cluster is assigned the *polarity* of the feature. To do that, MudRecS (i) generates a set of (cluster) *labels*, which are non-stop/-numerical/-sentiment named entities, that capture the features exhibited in the set of 99 reviews, (ii) assigns each sentence S in R to a feature cluster C based on the *similarity* between S and the label of C, and (iii) identifies positive/negative opinions on each feature using its clustered sentences.

Creating (Cluster) Labels for Identified Features

Using each review R in the set of 99 reviews on I, MudRecS creates concise and accurate *labels* that reflect the *features* specified in R using a suffix array algorithm, which has been proved to be efficient and effective in discovering key phrases in large text collections [20]. The suffix array algorithm generates a list of candidate (cluster) labels by simply extracting all the suffixes in the sentences in R. Since the generated list of suffixes may include non-representative labels (i.e., non-features) in describing I in R, MudRecS removes labels that are (i) *numerical* keywords, which seldom capture a feature of I, (ii) *incomplete*, i.e., included as substrings in other labels, (iii) *sentiment* keywords, i.e., words that express a positive or negative polarity, which is not considered as a feature of I, and (iv) non-named entities, which can be identified using the Stanford Entity Tagger (nlp.stanford.edu/software/CRF-NER.shtml), since features are expressed as named entities, such as *theme* and *plot*. In addition, labels that cross *sentence boundaries*, which indicate a topical shift, and end in the *Saxon genitive form* are excluded. Moreover, candidate (cluster) labels that reference web addresses are discarded and *stopwords* at the *beginning* or at the *end* of a (cluster) label are removed to enhance the readability of the label. Figure 2 shows a review on a movie extracted from Epinions.com with the *features* identified by MudRecS **bolded** in the review.

Assigning Sentences to Clusters

To cluster sentences in reviews on I which address the same (or similar) features of I, we have developed a simple, effective clustering method. Using the set of identified cluster labels CL and each retrieved review R on I, MudRecS computes the *degree of similarity* between each sentence S in R and label L in CL, denoted $LS\_Sim(S, L)$, as defined in Equation 3, using the word-correlation factors (introduced in Section 3.1.1). This process clusters sentences that address the same or similar feature F of I, which later allows MudRecS to determine whether F receives an overall positive/negative opinion captured in the set of 99 reviews on I by counting the total number of positive/negative sentences (based on sentiments in each sentence) assigned to F.

$$LS\_Sim(S, L) = \frac{\sum_{i=1}^{|S|} \sum_{j=1}^{|L|} Word\_Sim(w_i, w_j)}{|S|} \quad (3)$$

where $|S|$ ($|L|$, respectively) is the number of distinct, non-stop, stemmed words in S (L, respectively), w_i (w_j, respectively) is a non-stop, stemmed word in S (L, respectively), and $Word\_Sim(w_i, w_j)$ is the word-correlation factor of w_i and w_j. MudRecS normalizes $LS\_Sim(S, L)$ by dividing the accumulated word-correlation factors with the number of words in S to avoid favoring long sentences.

MudRecS computes $LS\_Sim(S, L)$ for each sentence S in each review R with respect to each label L in CL and assigns S to cluster C (labeled L_C) if $LS\_Sim(S, L_C)$ is the *highest* among all the labels in CL. Figure 3 shows some features (identified by their cluster labels) that are created using a set of reviews on the movie "Passions of the Christ" extracted from Epinions.com. Also displayed in Figure 3 are some sentences that are assigned by MudRecS to a feature.

Identifying Sentiment Opinions on Sentences

To determine the positive or negative polarity of a sentence S in a cluster, MudRecS calculates the *sentiment score* of S by subtracting the *sum* of the *negative* sentiment word scores in S (determined using SentiWordNet[2]) from the *sum* of its *positive* sentiment word scores in S, which reflects the sentiment of S such that if its sentiment score is *positive* (*negative*, respectively), then S is labeled as *positive* (*negative*, respectively).

3.2.3 Assigning Scores to Features

MudRecS computes a score for each feature of a rated item in two steps: *Review Analysis*, which is conducted once, and *Review Score Computation*, which is applied to each item considered for rating prediction and recommendation.

Review Analysis. For each item I in a given set S of multimedia items of the same domain previously rated at a rating level R_i ($1 \leq i \leq N$) by a user U, MudRecS determines which identified feature F in S receives an overall positive (negative, respectively) feedback based on the

[2] A SentiWordNet (sentiwordnet.isti.cnr.it/search.php) score of a word is a numerical value that indicates the *positive*, *neutral*, or *negative* orientation of the word.

Movies		Features							
		Story		Plot		Visuals		Direct	
Rating Level	# of Movies	+	-	+	-	+	-	+	-
1	5	2	3	1	4	3	2	0	5
2	10	3	7	4	6	5	5	2	8
3	20	8	12	7	13	6	14	13	7
4	25	18	7	10	15	7	18	13	12
5	15	11	4	9	6	4	11	12	3
Total	75	**42**	33	31	*44*	25	*50*	**40**	35

Table 2: The ratings of 75 movies provided by a user and a number of features of the movies with positive/negative polarity at each rating level

set of 99 reviews retrieved for each I by counting the total number of *positive* and *negative* sentences in I clustered under the label of F (that represents F). If F is assigned a *larger* number of *positive* (*negative*, respectively) than *negative* (*positive*, respectively) sentiment sentences clustered by MudRecS, then the polarity of F is treated as *positive* (*negative*, respectively). Hereafter, MudRecS counts the number of times F in items of S has been assigned a positive (negative, respectively) sentiment at each rating level, which constitutes the nominator in Equation 4. (See, in Table 2, the number of polarity of the feature *Story* at each rating level.)

MudRecS computes a *Polarity Feature Score*, denoted PFS, for each feature F with polarity P_F, which is either positive ('+') or negative ('-'), at each rating level R_i ($1 \leq i \leq N$).

$$PFS(R_i, F, P_F) = \frac{\#\ of\ Items\ in\ S\ with\ F\ and\ P_F\ Rated\ at\ R_i\ by\ U}{Total\ \#\ of\ Items\ in\ S\ Rated\ at\ R_i\ by\ U} \quad (4)$$

Example 1. Shown in Table 2 are the 75 movies rated at five different levels. The provided ratings on the 75 movies given by a user U were extracted from the Netflix dataset. Regardless of its rating level, MudRecS extracts 99 reviews retrieved for each one of the 75 movies M and determines the *polarity* of each *feature* covered in M. Table 2 shows the number of movies with an identified *feature*[3] that has been assigned a positive/negative polarity at each rating level.

The sentiments of each feature shown in Table 2 indicate that U is more interested in the $Story$ and $Direct(ion)$ of a movie, and less about $Plot$ and $Visuals$. □

Review Score Computation. Using the items ranked by U in the Review Analysis step, the Review score (RwS) of an unrated item I is computed at each rating level R_i ($1 \leq i \leq N$) by averaging the PFS of each feature F in I with polarity P_F, which is either *Positive* or *Negative*, at R_i.

$$RwS(R_i, I) = \frac{\sum_{F \in Features} PFS(R_i, F, P_F)}{N} \quad (5)$$

where $Features$ is the set of features identified in I through the 99 reviews retrieved for I, N is the total number of features (in $Features$) ranked at R_i as computed in the Review Analysis step, and PFS is as defined in Equation 4.

[3]MudRecS has identified 11 features for the set of 75 movies, but only 4 out of 11 are displayed, since they are features of the unrated movie "War of the World" to be evaluated.

Features: *S*(tory), *P*(lot), *V*(isuals), *D*(irection)
P_{Story} = '+', P_{Plot} = '-', $P_{Visuals}$ = '-', $P_{Direction}$ = '+'
PFS(1, *S*, +) = 2/5, *PFS*(1, *P*, -) = 4/5, *PFS*(1, *V*, -) = 2/5, *PFS*(1, *D*, +) = 0/5
PFS(2, *S*, +) = 3/10, *PFS*(2, *P*, -) = 6/10, *PFS*(2, *V*, -) = 5/10, *PFS*(2, *D*, +) = 2/10
PFS(3, *S*, +) = 8/20, *PFS*(3, *P*, -) = 13/20, *PFS*(3, *V*, -) = 14/20, *PFS*(3, *D*, +) = 13/20
PFS(4, *S*, +) = 18/25, *PFS*(4, *P*, -) = 15/25, *PFS*(4, *V*, -) = 18/25, *PFS*(4, *D*, +) = 13/25
PFS(5, *S*, +) = 11/15, *PFS*(5, *P*, -) = 6/15, *PFS*(5, *V*, -) = 11/15, *PFS*(5, *D*, +) = 12/15
RwS(1, "WoW") = 2/5 = <u>0.4</u>, *RwS*(2, "WoW") = 4/10 = <u>0.4</u>, *RwS*(3, "WoW") = 12/20 = <u>0.6</u>
RwS(4, "WoW") = 16/25 = <u>0.64</u>, *RwS*(5, "WoW") = 10/15 = **0.67**

Figure 4: The *Review score* (RwS) at each rating level of the movie "War of the Worlds" (WoW) computed using PFSs based on the polarities in Table 2

Example 2. The first two lines in Figure 4 show the features and their polarities identified by using the 99 reviews extracted for the unrated movie "War of the Worlds" (WoW). The RwS of WoW is the highest at rating level 5, which indicates that the (positive) features, i.e., *Story* and *Direction*, of WoW are regarded highly by U, which are reflected at the high levels of ranking (levels 4 and 5) of the two features as shown in Table 2. □

3.3 The Role Players of Multimedia Items

While some web users might (not) be interested in an item because of its genre(s) and/or particular features, others (dis)like the item because of its role player(s). For example, a user might enjoy movies acted by *Will Smith* regardless of the genre or visual effects of anyone of his movies. For this reason, besides analyzing the genres and reviews of multimedia items, MudRecS considers their role players.

A movie recommendation system likely suggests movies to a user U with the same actor/actress/director/producer who is highly rated by U, whereas a book recommender would suggest different books with the same authors (e.g., books published as a sequel). MudRecS uses information of role players in rating prediction by analyzing the *popularity* of a role player based on the ratings of the role player extracted from different multimedia websites (in Section 3.3.1). Hereafter, at each potential rating level of an unrated item I, MudRecS computes a *popularity* score for the role players of I, denoted $RolePlayerScore$ (RPS) (in Section 3.3.2).

3.3.1 The Popularity of Role Players

Popularity plays an important role in rating prediction because most users are influenced by the opinions expressed by others or the degree of exposition about an author/role player in the market [1]. The *popularity* of a role player P can be determined using various measures, such as the number of sold items involving P or user reviews, which capture the *popularity* of P[4]. Each popularity ranking is a numerical value between 0 and 1 such that if P is rated (in the ascending order) n out of m role players, then the ranking of P is $\frac{n}{m}$, such that the *lower* $\frac{n}{m}$ is, the *more popular* P is. The Web is rich with lists of rated role players for different types of multimedia data.

Movies. MudRecS relies on imdb.com to determine the popularity of role players in movies. Imdb.com creates ratings of movie actors, actresses, directors, and producers. The ratings provide a snapshot of who is *popular* based on the activities involving millions of imdb users on the website, which include the number of times an actor's/actress' page is

[4]The *Popularity* of a role player P is the by-product of the unique characteristics of P, which appeal to users [1].

viewed and *statistical data* such as movie ratings, box-office gross, the number of movies a role player involved, salaries of the role players, etc. These ratings are accessible through a service, known as STARmeter, on imdb.com, which reflect the popularity of the role players in movies.

Music. To determine the popularity of music artists, MudRecS considers the ratings on music artists provided by Billboard.com and Mtvmusicmeter.com. Billboard.com, which is a premier music website and a primary source of information on trends and innovation in music, archives an extensive array of (reviews on) songs and artists. The website *rates* artists based on *users' reviews* and the number of users who *favor* an artist A (through a clickable "favorite" button on the website). Billboard.com, however, ignores the *statistical information* on A, such as the sales records of A and the number of played/downloaded streams on his/her songs. For this reason, besides the ratings on and the (un)favorability of music artists provided by Billboard.com, MudRecS also considers Mtvmusicmeter.com to determine the rating of a music artist. Mtvmusicmeter.com is a website that offers access to archived profiles of over one million music artists and allows its users to browse/play their songs. The website establishes an *artist meter* that *rates* artists based on their popularity, which is determined by tweets, blog posts, news articles, streams, and sales records. To obtain a unified popularity score for a music artist A, MudRecS averages evenly the ratings of A provided by the two music websites.

Books. MudRecS defines the ratings of book authors using the authors' popularity ratings archived at iblist.com, which is an online database that boasts entries of over 19,000 authors. Iblist.com employs a *rating* system for authors similar to the ratings of actors/actresses/directors/producers on imdb.com. The website is managed by a board of volunteer editors and administrators who ensure the quality and accuracy of posted information. To avoid spam, irrelevant, and irresponsible remarks, third-party reviews are not accepted by iblist.com. Instead, only reviews submitted by registered, well-established users, who have been approved by the board of editors and administrators, can be posted on the website.

Paintings. Since there is no well-established or well-known website that provides a rated list of popular painting artists, MudRecS defines their ratings using Wikipedia. The website maintains a list of the highest, known prices for paintings, which MudRecS uses for identifying the popularity of painters such that the *higher* the cumulative price of a set of paintings created by a painter P is, the *higher* P is *ranked*. This notion is adopted, since the prices of paintings are primarily based on the artists' popularity and vice versa.

3.3.2 Assigning Scores to Role Players

MudRecS computes the *Role Player Score* (RPS) for each role player of a multimedia item in two steps: Role Player Analysis (performed once) and Popularity Computation.

Role Player Analysis. Given a set of multimedia items S of the same domain previously rated by a user and the *popularity* of each role player in S determined in Section 3.3.1, MudRecS computes a score for S, denoted *Popularity Score* (PoS), at each rating level R_i ($1 \leq i \leq N$), which measures the average popularity of all the role players in S at R_i.

$$PoS(R_i, S) = \frac{\sum_{I \in S} \sum_{A \in I} Popularity\ of\ Role\ Player\ A\ at\ R_i}{Total\ Count\ of\ Role\ Players\ in\ Online\ Sources} \quad (6)$$

Set of Movies (*S*)	Overall Rating	*Actor* 1 Ranking	*Actor* 2 Ranking
Iron Man	5	5/400	16/400
Wife	2	300/400	350/400
Cars	4	60/400	40/400
Love	1	250/400	350/400
X Men	5	1/400	10/400
Prison	3	100/400	150/400

Number of distinct movie actors/actresses = 400

PoS(1, *S*) = (250/400 + 350/400) / 2 = 300/400
PoS(2, *S*) = (300/400 + 350/400) / 2 = 325/400
PoS(3, *S*) = (100/400 + 150/400) / 2 = 125/400
PoS(4, *S*) = (60/400 + 40/400) / 2 = 50/400
PoS(5, *S*) = (5/400 + 16/400 + 1/400 + 10/400) / 4 = 8/400

New, Unrated Movie *I*, "War of the Worlds"
Ranking of *Actor* 1 in *I*: 5/400, Ranking of *Actor* 2 in *I*: 15/400
Avg_Pop(*I*) = (5/400 + 15/400) / 2 = 10/400, *RPS*(1, *I*) = 1-|*Avg_Pop*(*I*)-*PoS*(1,*S*)| = 110/400
RPS(2, *I*) = 1-|*Avg_Pop*(*I*)-*PoS*(2,*S*)| = 85/400, *RPS*(3, *I*) = 1-|*Avg_Pop*(*I*)-*PoS*(3,*S*)| = 285/400
RPS(4, *I*) = 1-|*Avg_Pop*(*I*)-*PoS*(4,*S*)| = 360/400, *RPS*(5, *I*) = 1-|*Avg_Pop*(*I*)-*PoS*(5,*S*)| = **398/400**

Figure 5: The computed RPS score at each rating level for the movie "War of the Worlds", where Actor i ($1 \leq i \leq 2$) in different movies can be different

EXAMPLE 3. Figure 5 shows the ratings provided by a user U on six movies extracted from the Netflix dataset and the rankings of two main actors/actresses in each movie at various rating levels obtained from STARmeter on imdb.com[5]. $PoS(5, S) = 8/400$, the lowest value among all the rating levels, indicates that U favors *popular* actors/actresses. □

Popularity Computation. During the recommendation process, MudRecS sums the popularity score of each role player of an unrated item I. The popularity score of each role player in I is then averaged into one score, denoted Avg_Pop, using Equation 7, and MudRecS assigns the RPS to I at each rating level in Equation 8.

$$Avg\_Pop(I) = \frac{\sum_{A \in I} Popularity\ of\ A}{Total\ Count\ of\ Role\ Players\ in\ I} \quad (7)$$

$$RPS(R_i, I) = 1 - |Avg\_Pop(I) - PoS(R_i, S)| \quad (8)$$

where PoS is as defined in Equation 6.

Equation 8 computes the difference between the average popularity of role players of I and the average popularity of role players of items rated at R_i by U. A *larger* RPS at R_i indicates that the popularity ranking of the role players of I is *closer* to the *popularity ranking* of role players of items previously rated at R_i, which increases the likelihood of assigning the rating level R_i to I. Equation 8 is complemented, since RPS is combined with other scores using the SCF (see Section 3), and the *higher* each *score* is at a rating level, the more likely the *rating* level is assigned to I.

EXAMPLE 4. Figure 5 shows the RPS at each rating level for the movie I, "War of the Worlds," which is played by popular actors as reflected by the averaged *Popularity Score* of role players of the movie, i.e., $Avg\_Pop(I) = 10/400$. The RPS for the movie is the *highest* at rating level 5, which increases the likelihood of the movie being rated high. □

3.4 Readability

The readability level of a book is a value between 0 and 13, i.e., Kindergarten to College. Reading can be a frustrating experience to users who struggle to understand (are not motivated to read, respectively) a book that is beyond (below, respectively) their readability levels. Analyzing the readability levels of books pre-rated by a user U can enhance

[5]An actor's/actress' ranking is the same in different movies.

Book	A	B	C	D	E	F	G	H	I	J
Reading Level (L)	11	3	9	3	12	7	11	11	3	11
Rating Level (R_i)	5	2	4	1	5	3	2	3	5	4

Unrated Book (I), "War of the Worlds"
Reading Level (L) = 11

ReS(1, *I*, 11) = 0/1 x (0/1 x 1/11 + 0/1 x 1/10 + 0/1 x 1/9 + 1/1 x 1/8 + ... + 0/1 x 1/2) = 0
ReS(2, *I*, 11) = 1/2 x (0/2 x 1/11 + 0/2 x 1/10 + 0/2 x 1/9 + 1/2 x 1/8 + ... + 0/2 x 1/2) = 1/32
ReS(3, *I*, 11) = 1/2 x (0/2 x 1/11 + 0/2 x 1/10 + ... + 1/2 x 1/4 + ... + 0/2 x 1/2) = 1/16
ReS(4, *I*, 11) = 1/2 x (0/2 x 1/11 + 0/2 x 1/10 + ... + 1/2 x 1/2 + ... + 0/2 x 1/2) = **1/8**
ReS(5, *I*, 11) = 1/3 x (0/3 x 1/11 + ... + 1/3 x 1/8 + ... + 1/3 x 1/1 + 0/3 x 1/2) = 1/216

Figure 6: The computed ReS value at each rating level for the book "War of the Worlds"

the accuracy of rating prediction on books unrated by U. MudRecS, which determines whether a book B unrated by U is appropriate for U based on the readability level of B, relies on the readability levels of books pre-rated by U.

Majority of existing readability assessment tools examine lexical, syntactic, and semantic content of a text document I to determine the readability level of I. These properties, however, are not well-designed and mostly based on observations. MudRecS relies on ReadAid [15], a fully-automated readability analyzer, to access the readability level of I.

Using a set of books S previously rated by a user U with their readability levels computed by ReadAid, MudRecS assigns a *readability score*, denoted ReS, at each rating level $R_i (1 \leq i \leq N)$ to a book I unrated by U based on the readability level L $(0 \leq L \leq 13)$ of I determined by ReadAid.

$$ReS(R_i, I, L) = \frac{Bks@R_iL}{TBks@R_i} \times \sum_{k=0, k \neq L}^{13} \frac{Bks@R_ik}{TBks@R_i} \times \frac{1}{|k-L|} \tag{9}$$

where $Bks@R_iL$ ($Bks@R_ik$, respectively) is the total number of books in S at readability level L (k, respectively) rated at R_i by U and $TBks@R_i$ is the number of books in S rated at R_i by U. The constraint "$k \neq L$" avoids a division by zero. $ReS(R_i, I, L) = 0$, if $TBks@R_i = 0$.

Equation 9 assigns a higher ReS to a book I at R_i if books in S with the same reading level L as I have been rated at R_i more frequently than at any other rating levels. $\frac{1}{|k-L|}$ assigns *higher* weight to similar, but not identical, grade levels, since users in grade levels L and $L+1$, or L and $L-1$, can usually comprehend the same materials, whereas users with grade levels 1 and 12, respectively cannot. The *smaller* the difference between two grade levels is, the *higher* it is *weighted* in a computed ReS, which is captured by $\frac{1}{|k-L|}$.

Example 5. Figure 6 shows the ratings on 10 books given by a BookCrossing user U. The rating on an unrated book I, "War of the Worlds," is determined by using the readability levels of the 10 books pre-rated by U. The readability and rating levels of the books show that books with *high* (*low*, respectively) readability levels are often ranked at *high* (*low*, respectively) rating levels 4-5 (1-2, respectively) by U. At the readability level of 11, I is most likely to be recommended to U, which is reflected by the ReSs, since ReS for I is the *highest* at rating level 4, indicating that the readability level of I is *closer* to the readability level of books in S rated at level 4 by U than at any other rating levels. □

3.5 Utilizations of MudRecS

MudRecS can be utilized differently to make recommendations. Each of the utilizations, listed below, can easily be implemented on top of MudRecS, which only requires a user interface to be constructed that handles users' inputs.

(i) When its users provide ratings on different multimedia items, MudRecS can automatically predict the ratings for other items of the same or a different domain in its underlying database, an approach currently adapted by Netflix.com.

(ii) When a user offers an item, which is either a book that the user has read, a song that (s)he has listened, a movie or painting the user has seen, or a rating on a multimedia item, MudRecS instinctively generates a number of recommendations (with the same or higher rating) on other items.

(iii) A user can specify a multimedia item I (s)he likes and a collection of multimedia items of different domains of interest. In response, MudRecS retrieves the items in the collection closed related to I with the highest rating.

(iv) Based on the multimedia items that a user U has previously accessed, MudRecS can suggest closely related items, which is similar to a shopping recommendation system, such as EBay.com and Amazon.com, that recommends items according to previous purchases.

MudRecS can predict the rating of either a movie, song, book, or painting I that is (not) in the same domain of items that a user U has previously rated using only the title of I.

4. PERFORMANCE EVALUATION

To assess MudRecS and compare its performance with existing state-of-the-art multimedia recommendation systems, we adopted the 5-fold cross-validation scheme, which is widely-used for evaluating recommendation and IR systems. Each dataset D (introduced in Section 4.1), which is created for the evaluation purpose, is randomly split into five equal-sized, disjoint subsets D_k $(1 \leq k \leq 5)$. For each D_k, MudRecS determines the genre, review, role player, and the readability (of a book) scores of each item in $\bigcup_{l=1, l \neq k}^{5} D_l$ and computes the Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE) (as defined in Section 4.2) on D_k. The RMSEs and MAEs of the five iterations are *averaged* to yield the corresponding mean error value, respectively.

We evaluate the rating predictions on unrated items determined by MudRecS and other recommendation systems on four distinct types of multimedia data: movies (videos), music (audios), books (text), and paintings (images). We compare MudRecS against four most-recently proposed multimedia recommendation systems (presented in Section 4.3), each of which predicts ratings on one (or more, but not all) of the four types of multimedia data. In fact, none of the existing recommendation systems, as we are aware of, predicts ratings on all four different types of multimedia data. Besides the four recommendation systems, we also compare the rating prediction accuracy of MudRecS on movies against 20 other movie recommendation systems participated in the Netflix contest (see details in Section 4.4). Even though we have evaluated MudRecS on text, audio, video, and image documents individually and independently, the evaluation of MudRecS can easily be extended to cover cross-domain multimedia items recommended for a user, assuming that the required genres, reviews, popularity of role players, and readability (for text data), are available.

4.1 Datasets

The datasets employed to evaluate (compare, respectively) MudRecS (with others, respectively) on rating prediction are well-known and widely-used, with the exception of paintings.

Figure 7: A snapshot of images in a Facebook survey conducted for constructing the paintings dataset

Movies. The MovieLens and Netflix datasets were chosen for evaluating the rating prediction accuracy of MudRecS on movies. The MovieLens (cs.umn.edu/Research/GroupsLens) dataset was created during a 7-month period from September 19, 1997 until April 22, 1998 by the developers of MovieLens, a popular web-based recommendation system on movies. The Netflix dataset, on the other hand, was collected as part of a contest in 2008 for predicting movie ratings.

Music. The Yahoo! Music Services dataset, which was created by the Yahoo! research team, contains ratings for songs collected from two different sources: the ratings provided by Yahoo! users of the Yahoo! music services in 2006, and the ratings on randomly selected songs collected during an online survey conducted by Yahoo! Research in 2006.

Books. The popular BookCrossing dataset [21] was manually created between August and September of 2004 with data extracted from BookCrossing.com.

Paintings. To the best of our knowledge, there is no benchmark dataset available for evaluating the performance of a recommendation system on paintings. To verify the accuracy of MudRecS in rating prediction on paintings, we randomly extracted 1,000 paintings on different genres (with reviews) using the Yahoo! image search engine. To obtain user ratings on the 1,000 paintings, we relied on Facebook users. We prepared 10 different Facebook surveys, each of which includes 100 distinct paintings. (A set of 100 paintings is an ideal collection, since a larger number of paintings in a survey would overwhelm its appraisers.) Each survey required each involved user to browse through the paintings and provide star ratings (on a 1-5 scale with '5' being the highest) on 10 or more paintings, using the rating meter directly below each painting. (Figure 7 shows a snapshot of the Facebook application that includes a number of paintings in one of the surveys.) The "Not Interested" button below each painting could be clicked by the user to assign a star rating of "1" (the lowest rating) to the painting. The surveys were sent out on May 4, 2011 to different Facebook users who were asked to forward the surveys to others. By May 11, 2011, we accumulated all the responses for our study.

Table 3 shows a summary of the multimedia datasets used for evaluating and comparing the performance of MudRecS.

4.2 Evaluation Metrics

Root Mean Square Error (RMSE) and *Mean Absolute Error* (MAE) are two performance metrics widely-used for evaluating rating predictions on multimedia data. Both

Multimedia Dataset	# of Users	# of Items	# of Ratings	Rating Scale
MovieLens	943	1,682	100,000	1-5
Netflix	480,000	18,000	100,000,000	1-5
Yahoo! Music	15,400	1,000	354,000	1-5
BookCrossing	278,858	271,379	1,149,780	1-10
FB Paintings	312	964	3,312	1-5

Table 3: Datasets used for evaluating MudRecS

RMSE and MAE measure the *average magnitude* of *error*, i.e., the average prediction error, on incorrectly assigned ratings. The error values computed by RMSE are squared before they are summed and averaged, which yield a relatively *high* weight to errors of *large* magnitude, whereas MAE is a *linear* score, i.e., the absolute values of individual differences in incorrect assignments are weighted equally in the average.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(f(x_i) - y_i)^2}{n}}, MAE = \frac{1}{n}\sum_{i=1}^{n}|f(x_i) - y_i| \tag{10}$$

where n is the total number of items with ratings to be evaluated, $f(x_i)$ is the rating predicted by a system on item x_i ($1 \leq i \leq n$), and y_i is an expert-assigned rating to x_i.

4.3 Recommendation Systems to be Compared

In this section, we detail the recommenders to be compared with MudRecS. These recommenders were chosen, since they achieve high accuracy in rating predictions on items in their respective multimedia domains.

MF. Yu et al. [19] and Singh et al. [17] predict ratings on books, movies, songs, and paintings[6] based on matrix factorization (MF), which clusters items and users by genres so that ratings are predicted according to the rating patterns of certain groups of items on certain genres. MF can also characterize both items and users by vectors of features, which are the properties of an item. The matrix with information on users, genres, and features of an item is decomposed by features and genres, which can be combined to provide rating predictions for any user-genre-feature combination.

Yu et al. propose a non-parametric matrix factorization (NPMF) method, which does not require the model dimensionality, i.e., users, features, and ratings, to be specified apriori. Rather, the dimensionality is determined from previously rated items instead. Singh et al. introduce a collective matrix factorization (CMF) approach based on relational learning, which predicts unknown values of a relation between entities of a certain item using a given database of entities and observed relations among entities.

ML. Besides the matrix factorization methods, probabilistic frameworks have been introduced for rating predictions. Shi et al. [16] apply a supervised machine learning (ML) approach to automatically construct a ranking model/function from training data to predict ratings on movies[7]. The ML approach applies a rank-oriented strategy, which

[6]The systems were originally designed to predict ratings on *books* and *movies* only but were implemented by us for comparing their predicted ratings on *songs* and *paintings* as well.
[7]The system was originally designed to predict ratings on *movies* but was implemented by us for additional comparisons on *books*, *songs*, and *paintings* as well.

exploits pairwise preferences between *items* and *users* to generate a list of ranked items corresponding to the ratings such that the *higher* an item is ranked, the *higher* its rating.

DM. Su et al. [18] propose uMender, a music recommendation system, which predicts ratings on songs by mining musical content and context information. uMender first utilizes the *perceptual patterns* of songs, which consist of the acoustical and temporal features of a song and then clusters users who provide similar ratings to similar songs to detect patterns for acoustics and temporal features, as well as to discover new, implicit, more applicable perceptual patterns. uMender assigns a rating to an item I based on the perceptual patterns of I and the cluster to which I belongs.

Netflix. We further compare MudRecS against the 20 systems that participated in the Netflix contest in 2008. The open competition was held by Netflix, an online DVD-rental service, and the Netflix Prize was awarded to the best recommendation algorithm with the lowest RMSE score in predicting user ratings on films based on previous ratings. On September 21, 2009, the grand prize of one million dollars were given. The RMSE scores achieved by each of the twenty systems, as well as detailed discussions on their rating prediction algorithms, can be found on the Netflix website (netflixprize.com//leaderboard).

4.4 Experimental Results

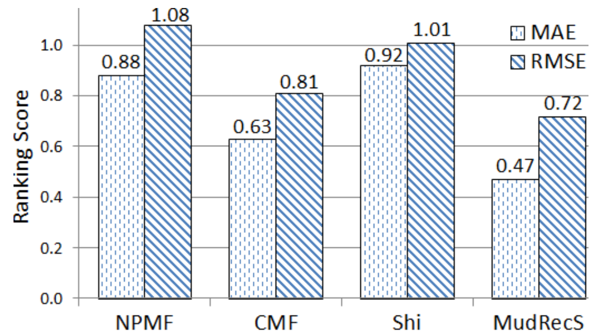
Figures 8(a), 8(b), and 8(c) show the MAE and RMSE scores of MudRecS and other recommendation systems on the MovieLens, Yahoo! Music, and BookCrossing datasets, respectively. As the RMSE scores indicate, MudRecS significantly outperforms other systems on rating predictions of the respective multimedia data. Regarding the MovieLens and Yahoo! Music datasets, MudRecS achieves a RMSE score of 0.72 and 0.80, respectively, which imply that, on the average, an incorrectly assigned rating by MudRecS is less than 1 star away from its correct assignment. On the BookCrossing dataset, MudRecS achieves an RMSE score of 0.45, which indicates that the error in rating assignment is less than half a star away from the user-/expert-assigned, actual rating. The MAE scores of MudRecS are also lower than the other systems on all the corresponding datasets. On BookCrossing, the MAE scores show that an error in rating predictions is on an average of 0.2 away from the correct prediction, which indicates a very high accuracy in rating predictions on books. Among all the four multimedia domains, the MAE scores of MudRecS are at least 0.16 lower than the other compared recommendation systems.

As shown in Figure 8(d) on *Paintings*, MudRecS achieves the highest RMSE and MAE values compared with other recommenders. Comparatively, the RMSE and MAE scores of MudRecS on paintings are the highest (i.e., worst) among the respective scores on other multimedia datasets. This is probably due to the smaller number of features of paintings in their reviews compared to other types of multimedia data.

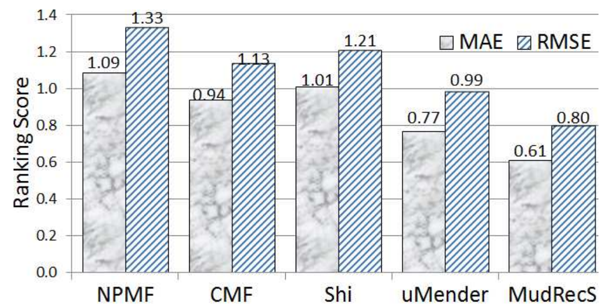
The small difference between the RMSE and MAE scores of MudRecS, as shown in Figure 8, verifies that the error in incorrect ratings predicted by MudRecS is insignificant, since the larger the errors is, the heavier they are penalized by the RMSE measure, which increases the difference between RMSE and MAE.

On the Netflix dataset, MudRecS achieves a RMSE score[8]

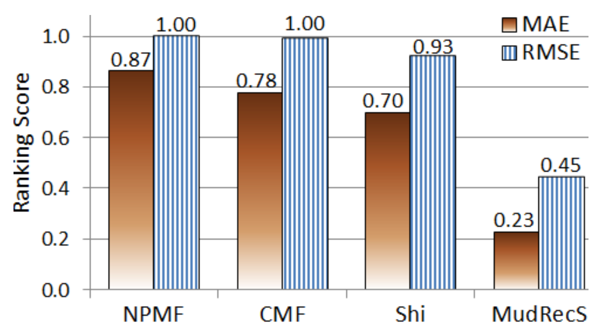
[8]MAE scores were not computed on the Netflix dataset due to their unavailability for the other 20 recommenders.



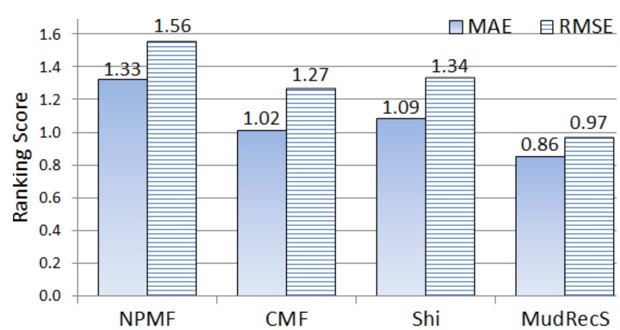
(a) The *MovieLens* dataset



(b) The *Yahoo*! *Music* dataset



(c) The *BookCrossing* dataset



(d) The *Painting* dataset

Figure 8: The MAE and RMSE scores for various recommendation systems on different datasets

of 0.8571. MudRecS outperforms 18 recommendation systems and is only outperformed by two systems (Belkor and Ensemble), both of which achieve the same score of 0.8567, a small, insignificant fraction (0.8571 - 0.8567 = 0.0004) better than MudRecS. The reason for the slightly better RMSE score achieved by the two systems on the Netflix dataset are twofold. Unlike MudRecS, Belkor and Ensemble were specifically designed for movie rating predictions, and the construction of their algorithms focus on rating patterns found in movies which may not apply to other domains. Moreover, Belkor and Ensemble account for temporal effects, i.e., the fact that a user's preference changes over time, which may lead to different ratings for the same movie over time. The temporal effect, however, does not apply to all users and requires a larger subset of training data in order to obtain reliable results, which are the constraints. In considering a 95% confidence interval, MudRecS significantly outperforms 17 recommendation systems and is *not* significantly outperformed by *any* of the twenty systems. CineMatch, Netflix's recommender, achieves an RMSE score of 0.9514 on the Netflix dataset, which is outperformed by MudRecS.

On the average, MudRecS takes approximately two seconds to make recommendations for a user query.

5. CONCLUSIONS

It is appealing to a user who is recommended multimedia items that are closely related to items that (s)he is interested in. For example, if a user U likes the movie "The Pursuit of Happyness," it is likely that U is also interested in "Seven Pounds," which shares the same genre (i.e., drama), actor (i.e., Will Smith), and similar reviews. Instead of matching the same type of multimedia items, the recommended items, which are either in the form of movies, music, books, and/or pictures, provide more sources of information for U. Offering different types of multimedia items enriches the learning and entertaining experience for the multimedia recommender users, which can be achieved by MudRecS, our proposed multimedia recommendation system.

Given the ratings of a set of multimedia items S provided by a user U, MudRecS predicts the *rating* of an item I unrated by U based on the *genres*, *role players*, *reviews*, and the *comprehensive level* of I. Offering different types of multimedia recommendations, which include movies (videos), songs (audios), books (text), and paintings (image) of interest to a user is the uniqueness and merit of MudRecS. MudRecS does not rely on user data extracted from social websites, ontologies, access patterns, nor training data, which is simple and easy to implement, since it depends only on users' ratings, genres, role players, reviews, and readability levels of multimedia items that are widely available on the Web.

Experimental results show that MudRecS accurately predicts the rating levels of multimedia items for recommendations. The empirical study demonstrates that MudRecS significantly outperforms existing multimedia recommendation systems in making recommendations based on ratings.

6. REFERENCES

[1] E. Arnould, L. Price, and G. Zinkhan. *Consumers*. McGraw Hill, 2002.

[2] L. Campos, J. Fernandez-Luna, J. Huete, and M. Rueda-Morales. Combining Content-Based and Collaborative Recommendations: A Hybrid Approach Based on Bayesian Networks. *Approximate Reasoning*, 51(7):785–799, 2010.

[3] H. Chen and A. Chen. A Music Recommendation System Based on Music Data Grouping and User Interests. In *ACM CIKM*, pages 231–238, 2010.

[4] C. Desrosiers and G. Karypis. *Handbook on Recommender Systems*. Springer, 2009.

[5] D. Dunlavy, D. O'Leary, J. Conroy, and J. Schlesinger. A System for Querying, Clustering, and Summarizing Documents. In *IPM*, pages 1588–1605, 2007.

[6] Q. Gu, J. Zhou, and C. Ding. Collaborative Filtering: Weighted Nonnegative Matrix Factorization Incorporating User and Item Graphs. In *SDM*, pages 199–210, 2010.

[7] Z. Guan, C. Wang, J. Bu, C. Chen, K. Yang, D. Cai, and X. He. Document Recommendation in Social Tagging Services. In *WWW*, pages 391–400, 2010.

[8] J. Koberstein and Y.-K. Ng. Using Word Clusters to Detect Similar Web Documents. In *KSEM*, pages 215–228, 2006.

[9] Y. Koren. Factorization Meets the Neighborhood: A Multifaceted Collaborative Filtering Model. In *ACM SIGKDD*, pages 426–434, 2008.

[10] O. Kwon. "I Know What You Need to Buy": Context-Aware Multimedia-based Recommendation System. *Expert Systems with Applications*, 25:387–400, 2003.

[11] G. Linden, B. Smith, and J. York. Amazon.com Recommendations: Item-to-Item Collaborative Filtering. In *Internet Computing*, pages 76–80, 2003.

[12] G. Luger. *Artificial Intelligence: Structures and Strategies for Complex Problem Solving,* 6^{th} *Ed.* Addison Wesley, 2009.

[13] T. Noh, S. Park, H. Yoon, S. Lee, and S. Park. An Automatic Translation of Tags for Multimedia Contents Using Folksonomy Networks. In *ACM SIGIR*, pages 492–499, 2009.

[14] Y. Park and K. Chang. Individual and Group Behavior-based Customer Profile Model for Personalized Product Recommendation. *Expert Systems with Applications*, 36(2):1932–1939, 2009.

[15] R. Qumsiyeh and Y.-K. Ng. ReadAid: A Robust and Fully-Automated Readability Assessment Tool. In *IEEE ICTAI*, pages 539–546, 2011.

[16] Y. Shi, M. Larson, and A. Hanjalic. List-wise Learning to Rank with Matrix Factorization for Collaborative Filtering. In *ACM RecSys*, pages 269–272, 2010.

[17] A. Singh and G. Gordon. Relational Learning via Collective Matrix Factorization. In *ACM SIGKDD*, pages 650–658, 2008.

[18] J. Su, H. Yeh, and P. Tseng. Music Recommendation Using Content and Context Information Mining. In *IEEE Intelligent Systems*, pages 16–26, 2010.

[19] K. Yu, S. Zhu, J. Lafferty, and Y. Gong. Fast Nonparametric Matrix Factorization for Large-Scale Collaborative Filtering. In *ACM SIGIR*, pages 211–218, 2009.

[20] D. Zhang and Y. Dong. Semantic, Hierarchical, Online Clustering of Web Search Results. In *ACM WIDM*, pages 69–78, 2001.

[21] C. Ziegler, S. McNee, J. Konstan, and G. Lausen. Improving Recommendation Lists Through Topic Diversification. In *WWW*, pages 22–32, 2005.