MovReC: A Personalized Movie Recommendation System for Children Based on Online Movie Features*

Yiu-Kai Ng

Computer Science Department
Brigham Young University
Provo, Utah 84602, U.S.A.

Email: ng@compsci.byu.edu

Abstract

- **Purpose**. Multimedia have significant impact on the social and psychological development of children who are often explored to inappropriate materials, including movies that are either accessible online or through other multimedia channels. Even though not all movies are bad, there are negative effects of offensive languages, violence, and sexuality as exhibited in movies. Parents and guidance of children need all the help they can get to promote the healthy use of movies these days.
- **Design/methodology/approach**. To offer parents appropriate movies of interest to their youths, we have developed MovReC, a personalized movie recommender for children, which is designed to provide educational and suitable entertaining opportunities for children. MovReC determines the appealingness of a movie for a particular user based on its children-appropriate score computed by using the Backpropagation model, pre-defined category using LDA, its predicted rating using matrix factorization, and sentiments based on its users' reviews, which along with its like/dislike count and genres, yield the features considered by MovReC. MovReC combines these features by using the CombMNZ model to rank and recommend movies.
- **Findings**. The performance evaluation of MovReC clearly demonstrates its effectiveness and its recommended movies are highly regarded by its users.
- **Originality/Value**. Unlike Amazon and other online movie recommendation systems, such as Common Sense Media, IMDb, and TasteKid, MovReC is unique, since to the best of our knowledge MovReC is the first *personalized children* movie recommender.

Keywords: Personalized recommendation, children, movie

1 Introduction

Web technology has been advancing at quantum speed since the Internet era with wearable technologies currently in the spotlight. The current generation, called the iGeneration [23], embraces technologies effortlessly and enjoys the conveniences that come with it. The iGeneration is changing our life style. We are constantly surrounded by TV's screen, smartphone's screen, smartwatch's screen, and game console's screen, to name a few. Adult's screen exposure is about 8 hours, or one third of, a day [28] while children,

*This article is a revised and expanded version of a paper entitled "Making Personalized Movie Recommendations for Children" published in iiWAS 2016 proceedings.

on average, spend 6 hours [31], i.e., one fourth of a day, or more staring at their devices' screen. Children's TV viewing habits have sifted from viewing tradition TV's set to on-demand services like YouTube and Netflix.

With the society embracing the iGeneration, reverting to less screen time seem improbable, and children's screen times will not subside but lengthen beyond the current 6 hours. Eventually, screen time will bleed into classroom time. The sheer hours of children's screen time come with a caveat, educational drop-off [22]. Is a child learning well at all during the iGeneration? Can a child understand geometry from playing games such as Angry Birds? Can children expose to important cognitive and social skills such as math and empathy from Netflix and online movies?

These days children age 2- to 4-year-old spend 63% of their screen time on education material, while only 32% of the screen time is considered educational for age 8- to 10-year-old [22]. According to Funk et al. [7], repeated exposure to movie and video game violence might cause children to develop a stronger proviolence attitudes. These attitudes often lead to more tolerance for violence and aggressive behavior, which proceeds to the development of desensitization toward violence [9]. Desensitization causes emotional numbness and creates a false sense of reality that violence is a norm, which promotes bad behavior with potential lurking social problem if preventive measures are not in place.

One of the most prevalent preventive measures adopted by the society to filter movies that might promote any undesirable behaviors such as violence, profanity, and sex is through the use of movie rating. Motion Picture Associate of America (MPAA), which represents the major Hollywood studios, acts as the de facto for the studios in forming and administering rating guidelines for the contents of films. MPAA's rating system consists of General Audiences (G), Parental Guidance Suggested (PG), Parents Strongly Cautioned (PG-13), Restricted (R), and Adults Only (NC-17) [6]. G-rated movies are generally appropriate for all ages, while PG might contain some material that are unsuitable for children. PG-13 has contents that only deem suitable for children 13 and over, and parents are advised to use discretion in deciding whether or not to let their children view it. R-rated movies contain some adult materials and are definitely unsuitable for children. NC-17 movies are only for adults, and children are not allowed any admission at all.

MPAA, at first glance, seem to do a decent job in providing parents, guidance, and educators with a stringent and consistent criterion to adhere to. However, Thompson and Yokota's findings claim otherwise [29]. High variability in movie ratings given by MPAA was found across a period of time. 51% (26 out of 51) of G-rated movies were found to contain tobacco, alcohol, or drugs, and 79% (1,007 out of 1,269) of PG-rated movies contain tobacco uses. Hence, it is up to parents, guidance, and educators in choosing the most appropriate movies for their children, while only using MPAA rating as a base guideline.

The outlook looks bleak if not for "transmedia storytelling," a term popularized by Henry Jenkins [13], which heightens character development, captures the imagination of original story plot, and makes it available via different media, including movies. According to the report in [1], children benefit immensely from transmedia and it might spark an interest in reading the original novel/play after watching the corresponding movie. The report further hypothesizes the integration of transmedia into the school curriculum, which can influence student's learning, in addition to social and ethical development. Clarke-Stewart and Beck [4] investigate the usage of video clips as part of maternal scaffolding's strategies to aid child's cognitive development, which has been proven to be effective. It has been shown that children are able to retell the story with desired details due to their mother's active involvement by discussing the characters' emotion, the intention of the plot, and many other critical aspects of the clip. Another study conducted by Ukpong et al. [30] shows that children role-model after their heroes and heroines through animated movies. They develop social norms and cues, besides picking up difficult vocabulary by emulating the fictional characters from the movies. Consequently, animated movies not only promote social skill development, but also sharpen cognitive skills. As such movies can be an efficacious education tool if put to right use and can enhance educational learning.

In view of the copious movies released each year, we have developed $MovReC$, a personalized movie recommendation system for children,[1] to aid parents/guidance in selecting appropriate movies of interest, whether for educational purpose or merely entertainment, for their children. MovReC tailors a user's needs by learning his interest based on the existing metadata, which include movie ratings, genres, personal ratings provided by either parents, guidance, or children, movie descriptions, movie reviews, and trailer's (dis)like counts. Using CombMNZ as a combination strategy [15] on feature scores computed using the metadata by applying the Backpropagation model for classification, the matrix factorization approach for predicted ratings, and the Latent Dirichlet Allocation (LDA) model [2] for topic analysis, MovReC automates what seems to be a laborious task of searching appealing movies and reading myriad movie reviews. As designed, MovReC not only acts as a time-saving system in filtering undesirable movies, it also gears towards the betterment of providing children with movies that can enhance their educational, social, and emotional development.

The remainder of this paper is organized as follows. In Section 2, we discuss the latest research work in movie recommendation systems. In Section 3, we detail the design methodology of MovReC. In Section 4, we present the empirical study conducted to evaluate the performance of MovReC and verify the preference expressed by end users on its suggestions. In Section 5, we give a conclusion.

2 Related Work

The engine behind a movie recommendation system is powered either by collaborative, content-based filtering, or a hybrid of the two. Collaborative filtering finds the nearest neighbors of a user U based on U's past rating history on similar movies' history, and attempts to infer the most probable recommendation list of movies for U. In the content-based approach, a user profile is built by obtaining the contents of movies rated by the user and the recommender will suggest a list of movies that matches the profile. Optimal recommendation is generally achieved when there are sufficient pre-existing ratings for the collaborative filtering approach to infer upon. However, collaborative filtering suffers from the *cold-start problem*. Choi et al. [3] propose using genre correlations to overcome the cold-start problem of movie recommendations. Their rationalization lies in movie's genres provided by movie experts, which are more dependable than users' ratings. MovReC considers both genres and users' ratings to harness the expertise of both the movie experts as well as the common consensus of the crowd.

Movie metadata, such as title, release date, genre, cast, and crew, are often used by content-based recommenders in suggesting movies that matches a user profile. This approach, however, does not often produce the best desirable results as claimed by Pilászy and Tikk [19]. This is likely due to a disparity between the descriptions as given by the movie studio over the actual movie itself. A comparison [19] was done between user ratings and movie's metadata, which include title, synopsis, actor, director, releases, and genre, and a conclusion was drawn: a mere use of 10 ratings from a movie outperformed the metadata in predicting user ratings. As stated before, MovReC considers both ratings and textual matadata of a movie, which compensate each other in making movie recommendations.

Yang et al. [33] attempt to tackle the cold-start problem by tapping into online social networks through the use of Bayesian-inference. The proposed methodology constructs Bayesian network based on the conditional probability distributions in measuring rating similarity among the user's friends through social networking. Although reported result looks promising on a devised social network topology, the feasibility of relying on a movie recommendation platform that contains both user's friend list and ratings have yet to be realized, since user anonymity was no longer valid. MovReC does not rely on a social network to predict ratings, but applies matrix factorization instead, which has been shown to be superior compared with the collaborative filtering approach, for rating predictions on movies.

[1] Children refer to youths under the age of 13 from here onward.

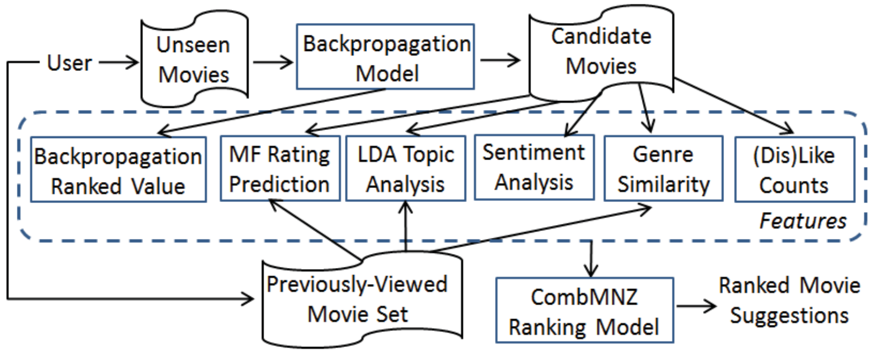


Figure 1: The children movie recommendation process of MovReC

A personalized movie recommender was introduced by Liu et al. [16] with the objective of providing users with a personalized movie synopsis or summary through users' preferences and movie contents. A list of candidate movies are pre-selected from weekly movie ranking information before any movie synopsis can be generated for user viewing. The generation of movie synopsis by the proposed recommender, however, implies that users are still required to invest time and efforts in analyzing the suggested movies to make a final decision on viewing the movies or not, which is avoided by a fully-automated movie recommendation system.

The mood-specific type of recommendation is a form of personalized recommendation that aims to incorporate context awareness into its recommender [26]. The recommender seeks to utilize movie-mood tags and generate a list of similar movies based on mood-to-mood similarity. A drawback of this approach is that if a movie comes without any mood tag, i.e., countering the cold-start problem, it will fall out of the potential list even though it might be appealing to a user.

Diao et al. [5] use topic modeling, users' ratings, and sentiments of the user reviews coupled with a statistical model as a base for their movie recommendation system. The recommender uses an inference algorithm to infer user's interest through the use of reviews against movie-specific words, such as characters in the movie. Even though MovReC also considers the topic, users' ratings, and sentiments of a movie, it can function as a movie recommender when anyone of these (inferred) metadata is missing, and thus is a more sophisticated recommender than Diao's. Moreover, MovReC tailors towards recommending movies to children, whereas the latter is not.

3 Our Recommender

In this section, we detail the design of MovReC which takes into account many features of a movie M to rank M among other movies for recommendation to a user. The metadata required to analyze each feature of M are widely available online which can be extracted from various websites. (See Section 4 for details.) The overall recommendation process of MovReC is shown in Figure 1.

3.1 Backpropagation Classification/Ranking

With a large number of movies released each year, which are in the thousands, it takes time and efforts to process all the movies and choose the ones that are appealing to an individual user. Given a particular user U, MovReC first selects movies, called *candidate movies*, that (i) have not been previously seen by U and (ii) are suitable for children. Non-children movies[2] are excluded by a trained backpropagation (BP) model, which classifies movies with contents involving violence, profanity, nudity, and gore. BP [17], which is a

[2] *Children movies* are either G- or PG-rated, whereas *non-children movies* come with other ratings.

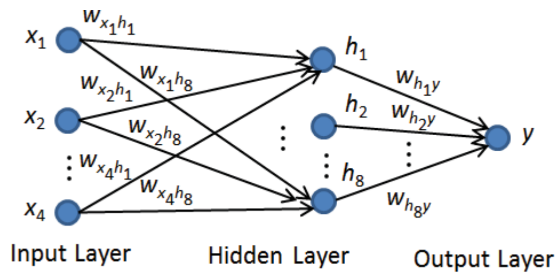


Figure 2: The trained backpropagation model, where $x_1, \ldots, x_4$ are input layer nodes, $h_1, \ldots, h_8$ are hidden layer nodes, y is the output layer node, and $w_{x_ih_j}$ (w_{h_jy}, respectively) is the weight between x_i and h_j (h_j and y, respectively), $1 \leq i \leq 4, 1 \leq j \leq 8$

supervised machine learning algorithm originally introduced in 1970s, is widely used for training artificial neural networks to perform categorization and ranking tasks [32].

A BP algorithm consists of input, hidden, and output layers. During its training process, a BP model learns a prediction function $y(x)$, where x is a set of input values and y categories x to one of the predefined classes and computes the ranking value of x using the *weights* across the model. The *hidden* layer is responsible for fine-tuning the *weights* assigned to different nodes of the BP through a number of iterations. Given each training instance I, the BP model adjusts/updates the weights associated with the corresponding pairs of (input to hidden and hidden to output, respectively) nodes based on the difference between the actual $y(I)$ and desired output of I, called *prediction error*. The error is propagated back to the input layer across the network through the hidden layer so that the weights of the edges could be adjusted accordingly during the next iteration. By repeatedly applying the propagation, the generated results of each iteration allow us to work backwards and compute the errors to be changed at each layer. Due to the responses of the errors propagated backwards during the iteration process, the algorithm is called *backpropagation*. The training process terminates when the *weights* of the edges become *stabilized*.

We trained a BP model for MovReC to classify (non-)children movies and compute their degrees of appropriateness for children. To train a BP model, we extracted 444 movies (un)suitable for children from Common Sense Media (commonsensemedia.org) as training instances, each of which is a movie with four *component* scores—Violence, Sex, Language, and Drinking/Drugs/Smoking.[3] The initial weight of each edge in the network was randomly assigned. Hereafter, the weights were repeatedly refined to progressively make the network more and more accurate, i.e., to minimize the errors in predicting the outputs, which are the category (1 for children and 0 for non-children) and ranking position computed by the sigmod function of each training instance. In the trained BP model for MovReC, the *input* layer consists of the *four* component nodes, whereas the *output* layer includes the classification/Ranking node. It is a common practice to maintain the number of hidden nodes approximately equal to twice the number of input nodes, and in the BP model employed by MovReC *eight* hidden nodes are included in its hidden layer, since we have empirically verified that using more or less hidden nodes we did not improve the performance of the model. (See the BP model used by MovReC as shown in Figure 2 in which $w_{x_ih_j}$ and w_{h_jy} ($1 \leq i \leq 4, 1 \leq j \leq 8$) are weights associated with input to hidden node and the hidden node to output node, respectively.) Using such function on (the component values of) a movie M, MovReC classifies and computes the corresponding BP ranking value for M.

We have chosen BP among other supervised machine learning models, since it is a widely-used, highly-effective function approximation method for pattern recognition and ranking features [35], which is used by MovReC for making children movie recommendations. In addition, BP achieves the same prediction

[3] Out of the 444 movies, 209 are G- and PG-rated movies, and the remaining 235 belong to other ratings. Another 226 (non-) children movies were used for testing purpose.

accuracy as other models, such as the support vector machines (SVM).[4]

Besides using the trained BP model to eliminate movies inappropriate for children, MovReC also excludes movies unseen by a user as candidate movies if "children" is not one of their genres which can be found in the MovieLens dataset (see Section 4.1 for details).

3.2 Rating Prediction

Making recommendations for users based on their past behaviors is crucial and is in essence learning hidden factors which drive users' decision-making process. Rating prediction is a classical approach for making movie recommendations, since the higher a predicted rating on a movie M for user U using the ratings of movies previously viewed by U is, the more likely M is appealed to U. Attempts have been made in the past by relating users to similar users and movies (an item) to similar movies (items) on which user-and item-based rating prediction systems are developed. A recommender implemented by utilizing the above ideas usually adapt the following design steps: (i) find the set of users most similar to the user U by matching U with users who share similar ratings on items, and (ii) find the items which receive the highest average ratings by the set of similar users. This recommendation strategy is intuitive, though it addresses the core of the problem in a rather indirect way, i.e., to reduce the problem of finding a user's decision latent-factor model to finding the set of users who make similar decisions. Matrix factorization is a sophisticated approach to use such a decision latent-factor model.

Given that there are f latent factors which together determine the item a user would likely prefer, in the context of movie recommendation, latent factors measure different dimensions such as seriousness versus escapism, comedy versus drama, and male- versus female-bias. This model represents each user ($p_u \in \mathbb{R}^f$) and item ($q_i \in \mathbb{R}^f$) as a vector containing a list of measures as dimensions. To find out a predicted rating on an item for a user, the inner product of the user and item vectors is computed, i.e.,

$$\hat{r}_{ui} = q_i^T p_u \tag{1}$$

where $q_i \in \mathbb{R}^f$ and $p_u \in \mathbb{R}^f$ are the item and user latent-factor vector, respectively.

To predict unknown ratings on movies, a recommender is given a $m \times n$ sparse matrix of known user-item ratings. Singular value decomposition (SVD) can be employed to deduce each user and item latent-factor vectors by factoring out the user and item latent-factor matrices from the user-item rating matrix. Traditional SVD, however, requires the given matrix to be dense. Assuming that all the missing entries are either zero or averages of other entries and applying classical SVD to fill the matrix is going to result in intolerable inaccuracy in the predictions. For example, the Netflix dataset consists of 100M ratings (in the range of 1 to 5) of 17K movies provided by 500K users with a total of 8.5 billion entries, out of which only 100 million of them are filled.[5] This extreme sparseness prompted researchers and computer scientists to look for modified solutions for SVD.

To find a function which maps a sparse user-item matrix with values representing ratings from a user to an item, to the user-item-latent-factor matrix with values represent latent-factor measure of each user or item, the singular values matrix S of a given matrix is utilized. S is the union of every feature vector q_i and p_u (as shown in Equation 1) such that the squared error of the inner products over the set of known user-item ratings is minimized, i.e.,

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

In Equation 2, K is the set of all user-item pairs for which the item rating is given, r_{ui} is the known rating, and $q_i^T p_u$ is the predicted rating. Equation 2 is a *cost function* for the Funk SVD Learning Algorithm, an

[4]We have also trained an SVM for ranking purpose which shows the same prediction accuracy.

[5]http://sifter.org/~simon/journal/20061211.html

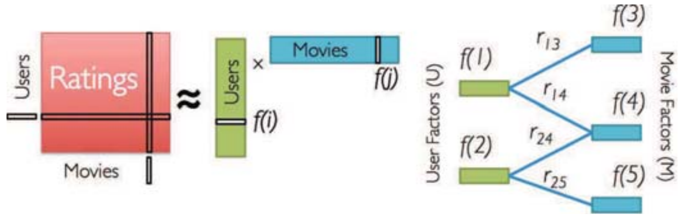


Figure 3: The matrix factorization of user and movie ratings

algorithm popularized by Simon Funk in an attempt to solve the Netflix 100M rating problem.[5] The basic idea is to employ techniques of gradient descent to iterate through the set of known ratings to minimize the squared error of the predicted rating. This iterative process involves the following steps: (i) before the training starts, a predicted rating was guessed to be the average item rating plus the user offset, (ii) for each given user-item rating, the prediction in the previous iteration is updated in the opposite direction of the gradient, and (iii) step (ii) was repeated until prediction error converges to zero. The following equations illustrate parameters involved more explicitly.

$$e_{ui} \stackrel{\text{def}}{=} r_{ui} - \hat{r_{ui}} \tag{3}$$
$$q_i \leftarrow q_i + \gamma(e_{ui}\, p_u - \lambda q_i) \tag{4}$$
$$p_u \leftarrow p_u + \gamma(e_{ui}\, q_i - \lambda p_u) \tag{5}$$

In Equation 3, e_{ui} is defined as the prediction error computed by subtracting the predicted rating $\hat{r}_{ui}$ from the known rating r_{ui}. Equations 4 and 5 state the prediction correction step of the current user and item latent-factor vectors, where γ is the learning rate, e_{ui} determines the direction and magnitude of the correction, and λ is the regularization strength.

In order to avoid overfitting of the data, punishment factors are added to Equation 2.

$$min_{q^*,p^*} \sum_{(u,i)\in K} (r_{ui} - q_i^T p_u)^2 + \lambda(||q_i||^2 + ||p_u||^2) \tag{6}$$

where λ controls the magnitude of regularization. Figure 3 captures the low-rank matrix factorization of user and movie ratings.

As we have mentioned before, a recommendation model based on matrix factorization requires that the set of hidden factors which drive users' decision-making process be learned. As each item and user feature vectors were induced directly from users' past ratings, which have been proven to prone to systematic bias, i.e., some items may consistently receive better/worse rating than others regardless of their feature vectors, some users may consistently give more/less critical ratings than the rest. In solving this problem, bias factors are introduced to ensure a fairer prediction. Equation 7 imposes the possibility of including user and item biases.

$$b_{ui} = \mu + b_i + b_u \tag{7}$$

where b_{ui} is the user-item bias accounting for the sum of bias factors not explained by the latent-factors model, μ is the global mean, b_i is the item bias of item i, and b_u is the user bias of user u. Since the predicted rating $\hat{r}_{ui}$ can now be restated as $b_{ui} + q_i^T p_u$, the cost function in Equation 6 can be rewritten as

$$min_{q^*,p^*} \sum_{(u,i)\in K} (r_{ui} - \mu - b_i - b_u - q_i^T p_u)^2 + \lambda(||q_i||^2 + ||p_u||^2 + b_i^2 + b_u^2) \tag{8}$$

An ancient Ring thought lost for centuries has been found, and through a strange twist in fate has been given to a small Hobbit named Frodo. When Gandalf discovers the Ring is in fact the One Ring of the Dark Lord Sauron, Frodo must make an epic quest to the Cracks of Doom in order to destroy it! However he does not go alone. He is joined by Gandalf, Legolas the elf, Gimli the Dwarf, Aragorn, Boromir and his three Hobbit friends Merry, Pippin and Samwise. Through mountains, snow, darkness, forests, rivers and plains, facing evil and danger at every corner the Fellowship of the Ring must go. Their quest to destroy the One Ring is the only hope for the end of the Dark Lords reign!

Figure 4: A description of the movie "The Lord of the Rings: The Fellowship of the Ring (2001)" posted under IMDb

where $\lambda(||q_i||^2 + ||p_u||^2 + b_i^2 + b_u^2)$ accounts for regularization to avoid overfitting, and the variable λ is a regularization strength constant which can be used to fine tune the training process.

3.3 Topic Analysis

Besides predicting users' ratings on children movies, MovReC also analyzes the *topic* covered in a candidate movie M to match the (same or similar) topics of movies previously viewed by a user. We utilizes the topic of M, since it offers a different dimension to measure the (dis)similarity between M and other movies. The Latent Dirichlet Allocation (LDA) model is widely used for discovering topics covered in text documents.[6]

LDA, or Latent Dirichlet Allocation, is a machine learning algorithm used in topic modeling. Given a corpus of movie descriptions made up of various words, which can be extracted from various movie websites, LDA identifies topic distribution throughout the corpus as well as topic distribution within a given movie description. This is accomplished by using words in the description to generate and assign topics. After an LDA has been successfully trained in generating useful topics from the corpus of movie descriptions, it can be adopted to identify the topic of a given movie. A study conducted by Jacobi et al. [11] has shown that LDA is effective at analyzing trends and patterns within a large digital news archive. The most useful aspect of LDA is that its application extends to all domains of information types. Figure 4 shows the description extracted from IMDb,[7] one of the world's most popular and authoritative sources for movie, TV, and celebrity content, on the movie "The Lord of the Rings: The Fellowship of the Ring (2001)".

The training process of an LDA begins with providing both a corpus of movie descriptions and the desired number of topics to create for the distribution. Given the inputs, the LDA algorithm determines each topic, which is represented by a set of keywords taken from the corpus, that spans the descriptions. This assignment of words to topics is done via sampling. After the LDA algorithm has successfully gone through multiple iterations for word reassignment, the end result is the generated latent topics present within the corpus of movie descriptions. These latent topics are made up of keywords assigned to each topic during the LDA training. It is important to note that while these topics are not automatically *labeled*, they can be manually created. Techniques such as *stopword removal* and *stemming* the descriptions before inputting them into the algorithm can help improve the quality of the results. Other factors that can affect the results are the number of topics chosen as input for the LDA and the number of descriptions in the corpus. By modifying these factors, one can improve the quality of the resultant latent topics.

To train an LDA model for MovReC, the inputs to the model include (i) a set of training instances, each of which is a movie description represented as a *sequence* of words, and (ii) the number of latent topics K to produce, which is 50 in our case.[8] LDA, which is a bag-of-word probability model involving

[6] In MovReC, documents used for training and processed by an LDA to determine the topic of a movie are *movie descriptions* extracted from movie websites.

[7] www.imdb.com/title/tt0120737/plotsummary

[8] We have empirically determined 50 as the number of latent topics to be considered which yields the best performance of

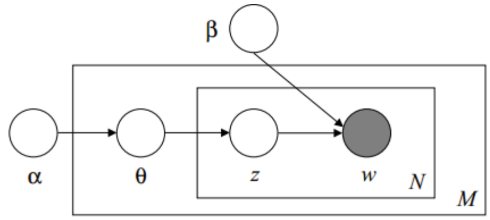


Figure 5: A graphical model representation of LDA in which the outer box represents documents, whereas the inner box represents the repeated choices of topics and keywords within a document

a corresponding generative process [2], estimates the *probability* of a word given a (latent) topic and the *probability* of a topic given a document. To determine the topic mixture for the document D based on the Dirichlet distribution over a predefined set of K topics, i.e., the *probability* of each word w in D belonged to each of the possible K topics, LDA (i) picks a topic for w according to the multinomial distribution that has been sampled, and (ii) generates w by using its corresponding topic based on the Gibbs sampling approach [8].

Intuitively, the generative process involves using LDA to learn the topic representation of each word w in D, i.e., the association of each word in D to each topic t. First of all, given t, LDA computes $p(t|D)$, which is the probability of words in D currently assigned to t, and $p(w|t)$, which is the probability of assigning w in all the documents to t. Hereafter, w can be reassigned to a new topic t such that $p(t|d) \times p(w|t)$ is the highest probability of t that generates w, which is the step to re-sampling the current topic of w. After a number of iterations, the assignment of each word to a topic is steady, which provides the estimation of words in documents associated to each topic based on the probability of each word assigned to each topic. We define LDA formally below.

In generative probability modeling, data arose from a generative process that includes hidden random variables. With *observed* and *hidden random variables*, the generative process defines a joint probability distribution. In LDA, the *observed* variables are words in the documents, whereas the *hidden* variables are the topics or topic structure. To infer the hidden topic structure from the documents, we simply compute the posterior distribution. The generative process for LDA can be defined using the following joint distribution of the hidden and observed variables.

$$p(\beta_{1:K}, \theta_{1:D}, z_{1:D}, w_{1:D}) = \prod_{i=1}^{K} p(\beta_i) \prod_{d=1}^{D} p(\theta_d) \prod_{n=1}^{N} p(z_{d,n}|\theta_d) p(w_{d,n}|\beta_{1:K}, z_{d,n}) \tag{9}$$

where $\beta_{1:K}$ are the topics where each β_k is a distribution over the given vocabulary, θ_d are the topic probability for document d, $\theta_{d,k}$ is the topic probability for topic k in d, z_d are the topic assignments for d, $z_{d,n}$ is the topic assignment for word n in d, w_d are the observed words for d, and $w_{d,n}$ is the word n in d.

Equation 9 specifies a number of dependent variables in which $z_{d,n}$ depends on per-document topic probability θ_d, and $w_{d,n}$ depends on z_d and all the topics $\beta_{1:K}$. Figure 5 depicts a graphical model representation of LDA, whereas Table 1 shows some of the topics used by MovReC and their corresponding keywords in the topic created by using LDA.

After an LDA has been trained, the classification process of LDA on a given movie V can be described as finding the probabilities of a number of topics covered in the description D of V and selecting the *topic* with the *highest probability* as the topic covered in D. After creating the latent topics based on Equation 9, Equation 10 is employed to determine the topic z of D based on the distribution of words in D and their probabilities in each topic z_j, i.e., $p\left(w_i|\, z_j\right)$, such that the topic z_j that has the *highest probability*, i.e., $p\left(z_j|\, D\right)$, is selected as the topic of D, i.e., topic of V.

MovReC in making children movie recommendations.

Table 1: Some of the topics created by using LDA and Gibbs sampling and their corresponding keywords

Topics	Keywords
Animal	Amuse, Bear, Bug, Cartoon, Chicken, Coyote, Disney, Dolphin, Duck, Park, Pig, Polar, Ranger, Spider, Stunt, Walt, Yogi
Family	Baby, Boy, Brother, Children, Daughter, Death, Father, Girl, Grandparent, Life, Mother, Old, Parent, Sister, Young
Musical	Band, Concert, Live, Music, Perform, Record, Rock, Show, Singer, Song, Stage, Star, Talent, Tour

$$Topic\_Match(V, D) = p\left(z_j \middle| D\right) = \max_{j=1}^{K} \sum_{i=1}^{M} p\left(w_i \middle| z_j\right) \tag{10}$$

where w_i is the i^{th} distinct word in D, z_j is the j^{th} latent topic among the K topics and is an input to LDA. $p\left(w_i \middle| z_j\right)$ is the *probability* of w_i in z_j, and M is the number of distinct words in D.

3.4 Sentiment Analysis and Opinion Mining

Sentiment analysis, often referred to as opinion mining, is a process of parsing natural language to quantify the opinion of a text; be it positive, negative, or neutral. The main purpose of sentiment analysis is to quickly get a close estimation of opinions. Good opinion mining will be correct about 70% of the time. At first this number does not seem that impressive, however, it is more impressive when one learns that when humans perform sentiment analysis they are only 'correct' 80% of the time. This is due largely to the subjective nature of the work. With this type of work there is not a right answer. Often, the measure of success is better achieved by looking at how often it tends to right. This process also has difficulties that do not yet have an effective solution. Opinion mining typically struggles with certain literary tools in natural language, like sarcasm. It is hard for a computer to understand context, and, therefore, it is hard to tell when a writer was being sarcastic. For similar reasons, it is difficult to distinguish homophones, words that have the spelling but different meanings. Misspellings or grammatical errors are difficult to decipher. Humans can typically perform a sentiment more accurately, however, computers can come close. It is also important to note that computers can perform a sentiment analysis extremely fast.

Because of the quick nature of computing sentiment analysis, it is utilized in a variety of different situations, particularly where speed is important or there is a lot of text to parse. One example is online social media. Social media faces the challenge to market advertisements (ads for short) to all varieties of people. Even though it may arguably have a slightly higher quality to have humans read and distribute targeted ads, it is much more efficient if computers do so. In fact, there is so much information on social media it is impractical for any person or group of people to review and understand how to best advertise to all its users. Through using a sentiment analysis, the computer can scan a person's posts and quantify how much the individual likes or dislikes the topics that they are posting about. Then using this information, the computer can suggest ads that are more in line with the individual's interest. Again, it's not a perfect science, but it is efficient and often very useful, especially when there is a lot of data. Another example of where a sentiment analysis might come in handy is to get a quick idea of public opinion. While President Obama was running for his second term he used this analysis to get very quick, almost immediate, feedback of public opinion on different policy changes or campaign decisions. Another usage is for different kinds of reviews. Sentiment analysis can quantify how good a new movie is or determine which cooking recipes are the most enjoyable.

There are several different ways to approach sentiment analysis. The different approaches to calculate sentiment analysis are usually broken up into three categories. These categories are *knowledge-based* techniques, *statistical* methods, and *hybrid* approaches. In this work with sentiment analysis we used a knowledge-based technique. The knowledge-based technique typically involves looking at each word and assigning it a positive or negative score and then averaging the sentiment score of the words together. Given a sentence, assign each word a quantitative score and then add them up and divide them by the total number of words. Typically, a *good* sentiment is a bit more complicated, but that is the general idea. The second kind of analysis is the statistical methods approach. The statistical methods use machine learning to mine the opinion using context. It attempts to use the underling grammar to decide whether the subject of the sentence has good things said about it or bad things. It can use its knowledge of the grammar of a particular sentence, in order to, account for double negatives. Again, it is giving a quantitative score just like in a knowledge-based technique. A hybrid method, as the name implies, attempts to combine the first two methods. The hybrid technique tries to optimize the advantages and minimize the disadvantages to each method.

MovReC analyzes the users' reviews on a candidate movie M to determine the ranking of M among other candidate movies. The analysis is a process of computationally identifying and categorizing (positive, negative, or neutral) opinions expressed in a set of reviews on M. Sentiment analysis is often used in natural language processing (NLP), computational linguistics (CLing), and intelligent text analysis (ITA) to classify and extract subjective opinions in source materials, which are users' reviews on M in MovReC. MovReC analyzes the *polarity* of reviews written by users who have watched M in the past and finds their opinions towards M. A highly-regarded comment on M indicates the corresponding user's positive and favorite attitude towards M, whereas a negative customer review on M reflects the dissatisfaction on either the performance of the director/actors/actresses, the plot, or the pace of M, among others. The analysis is based on the sentiment words, such as happy, bored, fantastic, and unimaginative, expressed in the reviews of M to draw a conclusion on the popularity/disapproval of M, which plays a role in recommending M.

MovReC conducts the sentiment analysis of movie reviews, which consists of four steps: parsing, eliminating stop words, extracting SentiWordNet value of each non-stop word, and averaging the total. After extracting a movie review, the text in the review is passed to the Stanford Parser. At this step, each word is assigned a part of speech. The Stanford Parser has a large range of parts of speech, including noun, verb, adjective, adverb, and others. Stopwords are first eliminated, since certain words like 'a' or 'the' are often used out of necessity, which do not add or detract from the true sentiment of a review, whereas non-stopwords are the necessary words used in both a positive or negative sentiment review. After eliminating the stopwords, each remaining word in a review is passed to the SentiWordNet with its corresponding part of speech from where a value that is a sentiment approximation for the word is returned. Each remaining word is averaged to obtain the sentiment score of the review. The conducted analysis indicates that if a movie viewer likes or dislikes a movie, the neutrality of the sentiment score would increase. The sentiment analysis on a movie is useful for several reasons. The analysis gives a qualitative score to each movie as seen by viewers in the past. This could potentially be used for recommending children possible movies to consider, which is valuable to other children who can quantify the quality toward potential movies to watch.

MovReC determines the polarity of each word w in a review R of a candidate movie M such that w is *positive* (*negative*, respectively) if its *positive* (*negative*, respectively) SentiWordNet[9] (sentiwordnet.isti.cnr.it) score is higher than its *negative* (*positive*, respectively) counterpart. Figure 6 shows the process of computing the sentiment score of a movie.

Example 1 Given below is a review on the movie, "The Lord of the Rings: The Return of the King", in which sentiment words are in **bold** along with their corresponding part of speech and sentiment score.

[9]SentiWordNet, a lexical resource for opinion mining, assigns to each word in WordNet three sentiment scores: positivity, objectivity (i.e., neutral), and negativity. A SentiWordNet score is bounded between -1 and 1, inclusively.

Figure 6: The process of computing the sentiment score of a movie (description)

> "In front of an amazingly **beautiful** (adjective, 0.83) scenery, Peter Jackson was able to **create** (verb, 0.06) a **fantasy** (adjective, 0.87) movie. Unlike so many others before did not deal with old cliches, it is far away from any **trash** (adjective, -0.42) movie a lot of people had expected it to be beforehand. Although I am sure that the cast of this film will soon be **forgotten** (verb, 0.05), it will be one of the most **renowned** (adjective, 0.44) pictures of the last decade."

In the second sentence, the sentiment analysis mistakenly thinks 'trash' as a negative word. This error is caused from variations in context. 'Trash' can be used to describe a difficult class, therefore, without context it is impossible to tell that 'trash' should have a positive sentiment. □

MovReC calculates the overall sentiment score of the users' reviews on M, denoted $Sti(M)$, by subtracting the sum of its *negative* words' scores from the sum of its *positive* words' scores, which reflects the sentiment orientation, i.e., positive, negative, or neutral, of the reviews on M. Since the length of the reviews on M can significantly affect the overall sentiment on M, i.e., the longer each review is, the more sentiment words are in the review, and thus the higher its (positive/negative) sentiment score is, MovReC normalizes the sentiment score of M by dividing the sum of the SentiWordNet scores of the words in the reviews by the number of sentiment words in the reviews on M, which yields $Sti(M)$ as defined in Equation 11.

$$Sti(M) = \sum_{i=1}^{n} \frac{\sum_{j=1}^{m} SentiWordNet(Word_{i,j})}{|Review_i|} \tag{11}$$

where n is the number of reviews on M, m is the number of words in the k^{th} ($1 \leq k \leq n$) review on M, $Word_{i,j}$ ($1 \leq i \leq n$, $1 \leq j \leq m$) is the j^{th} word in the i^{th} review of M, and $|Review_i|$ is the total number of sentiment words in the i^{th} review.

As the highest (lowest, respectively) SentiWordNet score of any word is 1 (-1, respectively), $LS < Sti(M) \leq HS$, where $-0.9 \leq HS \leq 1$, $-1 \leq LS \leq 0.9$, and $HS - LS = 0.1$. $Sti(M)$ is further scaled so that its value, denoted $Sti_{Scaled}(M)$, is bounded between 0 and 1, since a negative $Sti(M)$ value can be returned if the overall sentiment of M leans towards the negative region.

$$\begin{aligned} Sti_{Scaled}(M) &= CL(Sti(M)) + \frac{0.9 - FL(Sti(M))}{2} \\ CL(Sti(M)) &= \frac{\lceil Sti(M) \times 10 \rceil}{10}, \qquad FL(Sti(M)) = \frac{\lfloor Sti(M) \times 10 \rfloor}{10} \end{aligned} \tag{12}$$

3.5 Similarities in Genres

Movie goers rarely go to a movie without knowing what kind of movie they are going to watch, and each movie falls within a particular genre with a group of known audience. The *genre* of a movie M is differed from the *topic* of M, since the former indicates to which *type* M belongs, whereas the latter, which identifies the *theme* of M, addresses what M is about. A movie genre is a motion picture category, and movies that are similar in settings are grouped according to their genres. In fact, many movies are considered hybrids which are labeled by multiple genres. Commonly-used movie genres are shown in Figure 7 and considered by MovReC. We can distinguish the types of movies based on their genres.

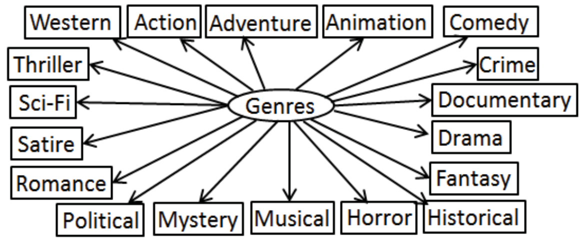


Figure 7: Genres considered by MovReC in suggesting movies

For example, adventure movies usually come with exciting stories with new experiences, whereas drama are plot-driven, portraying realistic characters and stories involving intense character development and interaction. MovReC compares the genres of a candidate movie and children movies watched by a user to make recommendations.

MovReC determines the *degree of similarity* between a candidate movie C and the set of children movies S previously viewed by a user using the *word-correlation factors* (wcf) [14][10] on each genre assigned to C against the *bag* of genres assigned to S. The similarity between C and S is computed in Equation 13 based on wcf.

$$Genre\_Sim(C, S) = \frac{\sum_{i=1}^{m} Min(1, \sum_{j=1}^{n} wcf(i, j))}{m} \tag{13}$$

where m (n, respectively) is the number of genres assigned to C (S, respectively), i (j, respectively) is a genre in C (S, respectively), and $wcf(i, j)$ is the *word-correlation factor* of i and j.

Using the $Genre\_Sim$ function, we *restrict* the highest possible similarity value between C and S to 1, which is the value for *exact* match. By imposing this constraint, we ensure that if C contains a genre k that is (i) an *exact* match of a genre in S, and (ii) similar to (some of) the other genres in S, then the degree of similarity of C with respect to S cannot be significantly impacted/affected by k to ensure a balanced similarity measure of C with respect to S.

3.6 Like and Dislike Counts

Existing movie websites, such as YouTube and Rotten Tomatoes, allow their users to give a thumbs up or down to a movie M by casting their *like* or *dislike* vote on M without writing a (brief) review on M. The counts summarize the (dis)approval/(dis)satisfaction opinions of the users on M, and indicate how well M is perceived by the users. These counts are considered by MovReC as a criterion in ranking M for movie recommendation, which is computed as

$$LCnt\_Ratio = \frac{Like\_Cnt - DLike\_Cnt}{Like\_Cnt + DLike\_Cnt} \tag{14}$$

where $LCnt\_Ratio$ is the ratio of subtracting the *dislike* count ($DLike\_Cnt$) from the *like* count ($Like\_Cnt$) over the total count, i.e., the sum of the *like* and *dislike* counts, of M.

3.7 CombMNZ

Based on the respective scores of the features computed for each candidate movie M in Sections 3.1 through 3.6, MovReC ranks all the candidate movies accordingly. To compute a single score on M which

[10]*A word-correlation factor* in the word-correlation matrix, which is a 54,625 $\times$ 54,625 symmetric matrix generated using a set of approximately 880,000 Wikipedia documents, indicates the *degree of similarity* of two words based on their (i) *frequency* of co-occurrence and (ii) relative *distance* in each Wikipedia document.

determines the cumulative effect of the six features for ranking propose, MovReC relies on the CombMNZ model [15].

CombMNZ is a well-established data fusion method for combining multiple ranked lists on an item I to determine a *joint* ranking of I, a task known as rank aggregation or data fusion.

$$CombMNZ_I = \sum_{c=1}^{N} I^c \times |I^c > 0| \tag{15}$$

where N is the number of ranked lists to be fused, i.e., the number of input ranked lists which is *six* in our case, I^c is the normalized score of I in the ranked list c, and $|I^c > 0|$ is the number of non-zero, normalized scores of I in the lists to be fused.

Prior to computing the ranking score of M, we transform the original scores in each feature ranked list of M into a *common range* [0, 1] by applying Equation 16.

$$I^c = \frac{S^I - I^c_{min}}{I^c_{max} - I^c_{min}} \tag{16}$$

where S^I is the score of I in the ranked list c to be normalized, I^c_{max} (I^c_{min}, respectively) is the maximum (minimum, respectively) score available in c, and I^c is the normalized score for I in c.

4 Experimental Results

In this section, we present the experimental results generated by the performance evaluation on MovReC. We first present the datasets used for training and testing different models adopted by various features discussed in Section 3. Hereafter, we introduce the appraisers who conducted the evaluation of MovReC.

4.1 Datasets

Even though there are datasets available for conducting performance evaluation of a movie recommendation system, there is no dataset to evaluate the performance of children movie recommenders. We used multiple datasets for training and testing the performance of different models adopted by different features and considered by MovReC. Besides using the Common Sense Media dataset for training a Backpropagation model (see details in Section 3.1) for classifying (non-)children movies, we have created other datasets. Note that the dataset used for training the BP model (LDA, respectively) is disjoint from its corresponding test dataset.

To the best of our knowledge, there is no benchmark dataset available for evaluating the performance of children movie recommenders. The most commonly-used movie dataset for performance evaluation of movie recommendation systems is the MovieLens dataset, and *MovieLens 20M* dataset [10] is the latest stable benchmark dataset.[11] MovieLens 20M contains 20M ratings (on the scale of 1 to 5) on 27,278 movies provided by 138,483 users between January 9, 2015 and March 31, 2015.

- *Rating Prediction*. We randomly chose 4,200 movies rated by 1,000 users from MovieLens 20M as the dataset for rating prediction, denoted *RMovLens*. RMovLens consists of 126,501 (raw) user ratings, and 4,750,500 predicted ratings were generated by using matrix factorization.
- *Topic Analysis*. To train the LDA for analyzing the topic of a movie by MovReC, we downloaded the descriptions of 8,306 movies from IMDb (Internet Movie Database), an online database of information related to movies, video games, and TV programs. IMDb archives movie summaries, reviews,

[11] http://files.grouplens.org/datasets/movielens/ml-20m-READ ME.html

plot, etc., and summaries are descriptions employed by MovReC for training the LDA and ranking candidate movies.

- *Similarity of Genres*. To determine the genres commonly-assigned to children movies, we extracted the genres of 1,184 children and 2,911 non-children movies from the MovieLens 20M dataset. A total of 4,095 movies were used as the test set for evaluating the performance of MovReC, denoted $TSet$, in terms of measuring genre similarity and ranking recommended children movies. $TSet$ is a subset of RMovLens.

- *Sentiment Analysis*. To analyze the sentiment of each test movie, we extracted the user reviews from IMDb for each one of the 4,095 movies in the $TSet$. There are between 0 and 1,000 user reviews for each one of the 4,095 movies.

- *(Dis)Like Counts*. To compute the LCnt_Ratio of each movie M in $TSet$, we used the *Like* and *Dislike* counts of M extracted from YouTube(.com). YouTube is a popular video-sharing website which allows users to upload movies. Due to the copyright constraint, recently-released movies are not available to be viewed by YouTube users. However, movie trailers and their (dis)like counts are widely accessible on YouTube, which are used by MovReC for computing the LCnt_Ratio of a movie in $TSet$.

Note that the set of candidate movies C to be considered for recommendations by MovReC to a user U and the set of movies previously viewed by U are disjoint subsets of RMovLens.

4.2 Mechanical Turk's Appraisers

Amazon Mechanical Turk (MT), an online crowdsourcing marketplace provided by Amazon Web Services, offers a large temporary, on-demand workforce and gives its workers a selection of thousands of tasks to complete at their own time. Employers of MT appraisers, called *requesters*, can create a Human Intelligence Task (HIT), which can be a question-answering survey, multiple-choice performance evaluation, or classification/clustering task, for MT appraisers, called *workers*, to complete. Each HIT created by us includes *three* sample children movies (identified by their titles) that have been viewed by a user U and a list of *nine* children movies (also identified by their titles) which are the top-9 movies recommended by MovReC based on the three sample movies and are posted in random order. For each HIT, each involved MT appraiser was required to pick *three* out of *nine* movies (in descending order) most-likely appealing to U. The appraiser is given an option not to choose any recommended movies.

We set up eight HITs to evaluate the performance of MovReC in terms of making recommendations on movies of interests to users who can be anyone, including parents/guidance of children or children themselves. Each of the eight HITs was created by randomly chosen a user from RMovLens who had viewed at least three children movies in the past. For each of the HITs, we received responses from *fifteen* MT appraisers between July 31, 2016 and August 12, 2016, with a total of 120 appraisers' evaluations.

4.3 Performance Metric

To determine the overall degree of appealing of a movie M recommended by MovReC, we applied a simple *counting scheme* on M. If M is marked as "should be recommended" by at least one of the MT appraisers who evaluated M, a point is rewarded to M. Based on this counting strategy, we determined the top-3 counts of recommended suggestions[12] made by the MT appraisers for each HIT, which yields the *gold standard* for performance evaluation.

[12] Top-3 recommended movies are considered, since highly appealing movies are more useful than marginally appealing movies. In fact, the lower the ranked position of an appealing movie, the less useful it is for a user, since it is less likely to be examined by the user.

We measure the degree of effectiveness in making useful movie suggestions by MovReC based on the *Normalized Discounted Cumulative Gain* (nDCG) [12] on the corresponding top-3 suggestions for each test case, i.e., HIT. nDCG, which penalizes useful suggestions that are ranked *lower* in the list of suggested movies, uses the *graded relevance value* as a measure of the usefulness, or gain, of a recommended movie. *Gain* is accumulated starting at the top of the ranking and may be reduced, or discounted, at lower ranks.

$$nDCG_3 = \frac{DCG_3}{IDCG_3}, DCG_3 = \sum_{i=1}^{3} \frac{2^{rel_i} - 1}{log_2(i+1)} \tag{17}$$

where rel_i is the *graded relevance* (1 or 0) of the movie at rank position i, $log_2\, i$ is the discount/reduction factor applied to the *gain*, and $ICDG$ is the ideal (perfect) Discounted Cumulative Gain.

4.4 Performance Evaluation of MovReC

To evaluate the effectiveness of MovReC in suggesting children movies of interests to its users, we gathered the responses made by the MT appraisers on the eight HITS (as discussed in Section 4.2) which set the *gold standard* on the movies "should be recommended" based on a set of movies previously viewed by a user.

4.4.1 The HIT Evaluations

Each of the HITs evaluated by its corresponding set of appraisers consists of the following information, and a sample HIT is shown in Figure 8:

(i) The *title* of each one of the three movies M previously viewed by a user, who is randomly chosen, is given. Along with the title, the first *sentence* in the description of M is also provided to the appraisers as a reference to the movie to illustrate the plot of M.

(ii) The *instructions* on making the 1^{st}, 2^{nd}, and 3^{rd} choices on the movies that "should be recommended" are given to the appraisers. Three *buttons* with labels '1', '2', and '3' are available for the appraisers to mark their choices on movies that "should be recommended."

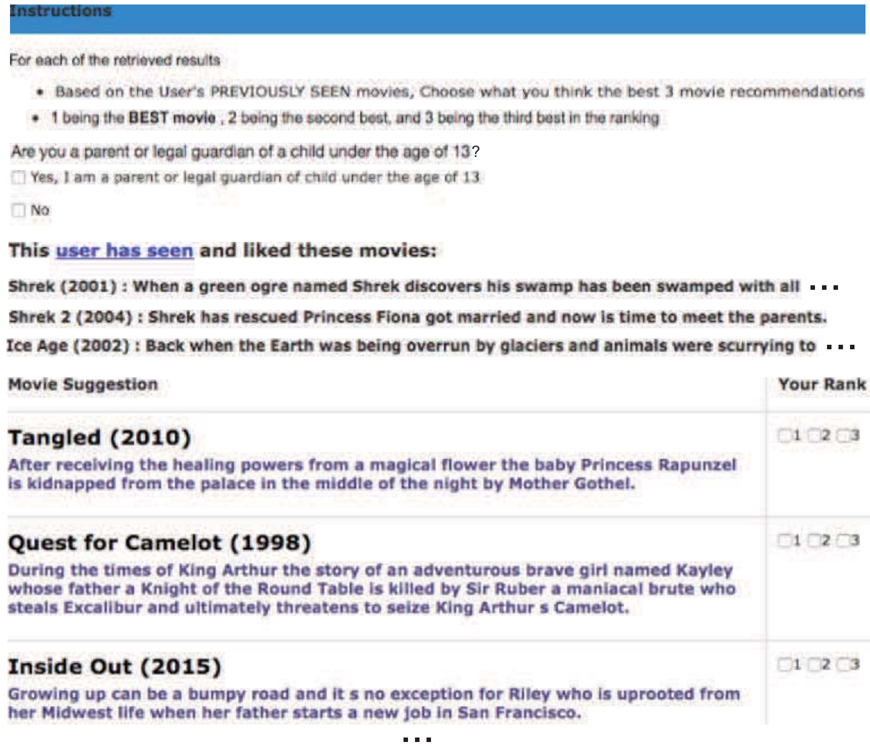
(iii) The top-9 movies *recommended* by MovReC for a particular HIT are shown and randomly ordered, each of which is identified by its title and the first sentence of its description.

(iv) Each appraiser was requested to indicate whether (s)he is a parent/guidance of children, which provides the background information of the appraisers for performance analysis purpose.

Table 2 shows two different sample sets of top-3 recommendations made by MovReC, and each of the recommendations is based on a set of previously-reviewed movies, respectively.

4.4.2 MovReC as a Movie Recommender

nDCG, as defined as $nDCG_n = \frac{DCG_n}{IDCG_n}$, where n is any natural number, is a ranking metric. In the field of information retrieval (IR), $IDCG_n$ produces the maximum possible DCG value of a ranked list of suggestions L through position n, which is often utilized as a normalized factor of L. For example, given a sorted list of 100 suggestions with 10 useful suggestions, the nDCG is 1 if all the useful suggestions are ranked in the top 10. Hence, if an nDCG value of a ranked list L is 0.8, then L achieves an 80% of the best ranking possible. Figure 9 depicts the nDCG value of each HIT and the average nDCG of the eight HITs achieved by MovReC. The average nDCG value indicates that useful suggestions made by MovReC often appear in the upper half of the corresponding ranked list.



Instructions

For each of the retrieved results

- Based on the User's PREVIOUSLY SEEN movies, Choose what you think the best 3 movie recommendations
- 1 being the **BEST movie** , 2 being the second best, and 3 being the third best in the ranking

Are you a parent or legal guardian of a child under the age of 13?

☐ Yes, I am a parent or legal guardian of child under the age of 13

☐ No

This user has seen and liked these movies:

Shrek (2001) : When a green ogre named Shrek discovers his swamp has been swamped with all ...

Shrek 2 (2004) : Shrek has rescued Princess Fiona got married and now is time to meet the parents.

Ice Age (2002) : Back when the Earth was being overrun by glaciers and animals were scurrying to ...

Movie Suggestion	Your Rank
Tangled (2010) After receiving the healing powers from a magical flower the baby Princess Rapunzel is kidnapped from the palace in the middle of the night by Mother Gothel.	☐1 ☐2 ☐3
Quest for Camelot (1998) During the times of King Arthur the story of an adventurous brave girl named Kayley whose father a Knight of the Round Table is killed by Sir Ruber a maniacal brute who steals Excalibur and ultimately threatens to seize King Arthur s Camelot.	☐1 ☐2 ☐3
Inside Out (2015) Growing up can be a bumpy road and it s no exception for Riley who is uprooted from her Midwest life when her father starts a new job in San Francisco.	☐1 ☐2 ☐3

...

Figure 8: The snapshot of a sample HIT used for evaluating the performance of MovReC

4.4.3 Top-Ranked Position Evaluation

As in many (web) search applications, users of recommendation systems tend to examine only the few top-ranked suggestions to find useful information. For this reason, the performance evaluation of a recommendation system focuses on the effectiveness measure of the system at making appealing suggestions at very high ranks, usually suggestions close to the top of the ranking. One of these measures is *precision* at rank position p, where p is typically 10 or less. In the performance evaluation of MovReC on making top-ranked, appealing suggestions, we computed the average *Precision*@1 (P@1), average *Precision*@3 (P@3) , and *Mean Reciprocal Rank* (MRR). Average P@1 measures the usefulness of the top-ranked suggestions made for the HITs, and average P@3 computes the ratio of the usefulness of the top-3 ranked suggestions of the HITs, whereas MRR calculates the average of the reciprocal ranks at which the *first* useful suggestion for each HIT is made. For example, if one of the top-5 movie suggestions made by MovReC for a particular user U is M_n, M_n, M_n, M_u, and M_u, where M_n denotes *not useful* and M_u denotes *useful*, the reciprocal rank is $\frac{1}{4}$ = 0.25. Further assume that the second set of top-5 movie suggestions made by MovReC for U is M_n, M_u, M_u, M_n, M_n, the reciprocal rank is $\frac{1}{2}$ = 0.5. Hence, the MRR for the two set of rankings is $\frac{(\frac{1}{4}+\frac{1}{2})}{2} = \frac{3}{8}$. Figure 10 shows that almost *two-third* of the 1^{st}-ranked movies suggested by MovReC are treated by appraisers as *useful* and *half* of the top-3 suggestions are regarded as useful by the appraisers. Moreover, the MRR value reflects that the useful suggestions made by MovReC often appear in the first quartile.

4.4.4 Individual Feature Evaluation

In making movie recommendations, MovReC considers different features, including *children-appropriate ranking* values computed by the trained BP model, movie *ratings predicted* by matrix factorization, movie *topics* analyzed by LDA, *sentiments* of movies determined by using movie reviews, *genre similarity* mea-

Table 2: Top-3 recommendations (identified by their titles) made by MovReC on two different sets of three movies previously-viewed by a user, where *gold standard* on the three movies determined by MT appraisers are *italicized*

HIT#	Previously-Viewed Movies	1^{st} Suggestion	2^{nd} Suggestion	3^{rd} Suggestion
4	(i) The Wizard of Oz (1939) (ii) Alice in Wonderland (1951) (iii) Peter Pan (1953)	*Tinker Bell (2008)*	A Bug's Life (1998)	*The Lion King* (1994)
6	(i) A Bug's Life (1998) (ii) Rango (2011) (iii) Finding Nemo (2003)	*Shrek 2 (2004)*	*Shrek (2001)*	*Once Upon a Forest* (1993)

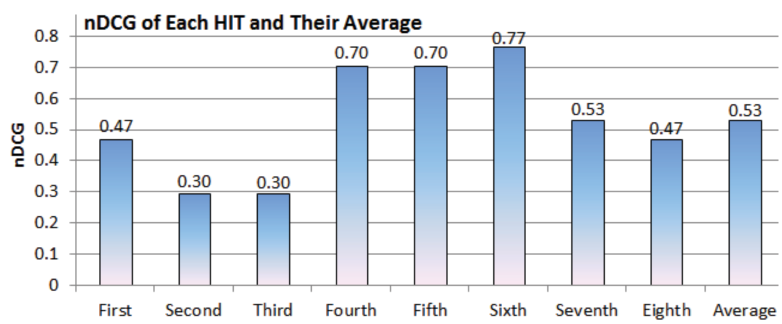


Figure 9: The nDCG value of each HIT and their average achieved by MovReC

sured by likelihood of different types of movies, and *(dis)like counts* based on the (dis)satisfaction of movie viewers on movies.

We have conducted an empirical study to verify that matrix factorization is the most accurate approach among the well-known and widely-used rating prediction models, which includes the user-based Collaborative Filtering (UCF) [25], item-based Collaborative Filtering (ICF) [21], and Expectation Maximization (EM) approach [24]. We briefly discuss each of these rating prediction approaches below.

- **User-based Collaborative Filtering (UCF)** methods predict the rating r_{u_i} of a user u for an item i using the ratings provided to i by users most similar to u, which is usually referred as nearest-neighbors (NN). Assume that for each user $v \neq u$, a value $w_{u,v}$ denotes the preference similarity between u and v. The k-nearest-neighbors (k-NN) of u, denoted by $N(u)$, are the k users v with the highest similarity $w_{u,v}$ to u. Note that only the users who have rated item i can be used in the prediction of r_{u_i}, and only the k users most similar to u that have rated i are considered. The rating r_{u_i} is estimated by averaging ratings given to i by these neighbors.

- **Item-based Collaborative Filtering (ICF)** approaches capture the fundamental relationships among different items such that two items are *similar* if the users agree about their ratings. The similarity of items can be captured by using a $m \times m$ matrix M such that the element $M_{i,j}$ denotes the similarity between items i and j. Item-based Collaborative Filtering algorithms represent items in the user-rating space such that an item is a vector whose dimensions are the ratings given by the n users and the coordinate of each dimension is the user rating. As a consequence, the relationships among items are expressed by means of the similarities among the related vectors.

- **Expectation Maximization (EM)** is a multi-criteria rating prediction method that requires the users to provide more data than their single-rating counterparts, and thus increasing the likelihood of ob-

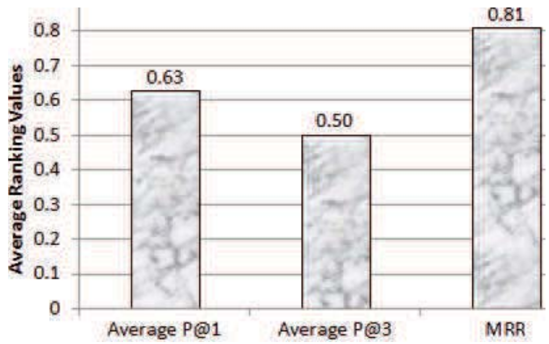


Figure 10: The average P@1, P@3, and MRR scores achieved by MovReC on the eight HITs

taining missing or incomplete data. A popular technique to deal with missing data is the Expectation Maximization (EM) algorithm that finds maximum likelihood estimates for incomplete data. In comparison, Collaborative Filtering (CF) techniques attempt to predict what information can meet a user's needs based on data coming from similar users, which requires only a single rating as input. The multi-criteria-based approach, on the other hand, presents a possibility to provide accurate prediction by considering the user preferences in multiple aspects and they can be an appropriate alternative choice [18].

To verify the rating prediction accuracy using matrix factorization as opposed to the most commonly-used rating prediction approaches listed above, we used the popular MovieLens dataset which was made available by GroupLens Research Group.[13] The full data set contains 24 million ratings applied to 40,000 movies that were provided by 260,000 users and were last updated on October 2016. Ratings are made on the scale of 5-star scale, with half a star increments, i.e., ratings are between 0.5 stars and 5.0 stars. In computing the prediction accuracy of each prediction model, the rating R on each movie that a user has provided was predicted by using each one of the models being considered individually, i.e., ICF, UCF, EM, and MF, without using R, and the predicted accuracy is computed by taking the absolute value of the difference between R and its predicted rating. Figure 11 clearly shows that the matrix factorization approach outperforms the other well-known rating prediction models. All of these results are statistically significant based on the Wilcoxon Signed-Ranks Test ($p < 0.001$).

In order to further justify the necessity of employing all of the six features adopted by MovReC for identifying and ranking appealing movies for a user, we have conducted an empirical study which analyzes the capability of each individual feature in making useful movie recommendations and compares its performance with MucReC which employs all the features. As shown in Figure 12, MucReC significantly outperforms each of the individual features in making useful suggestions to its users based on the gold standard determined by the MT appraisers using the HITs. The computed nDCG score of each feature, as well as the nDCG of MovReC which combines all the features using the CombMNZ model, clearly indicate that MovReC takes the advantage of the individual strength of each feature and greatly improves its effectiveness and the ranking of its suggested movies. The overall nDCG of MovReC (as shown in Figure 12), which is 0.53, is a statistically significant improvement ($p < 0.0001$) over the nDCG score achieved by any individual feature based on the Wilcoxon signed-ranked test.

4.4.5 General Appraisers Versus Parents/Guidance

We have gathered another statistical result on the performance evaluation of MovReC using the HITs based on parents/guidance and the general appraisers. Among all the MT appraisers who evaluated the

[13]https://grouplens.org/

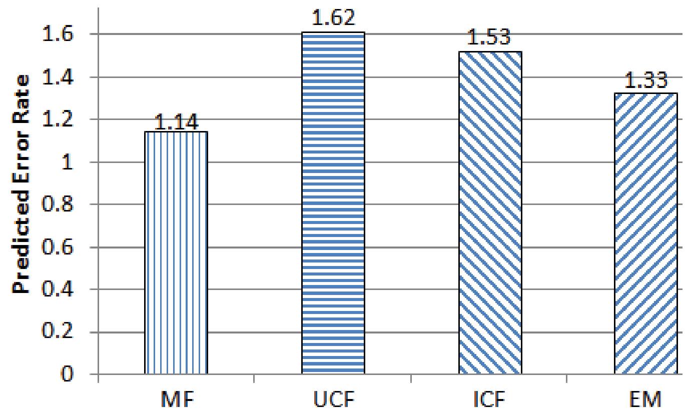


Figure 11: Rating prediction accuracy computed using the MovieLens dataset on the matrix factorization (MF), User-based Collaborative Filtering (UCF), Item-based Collaborative Filtering (ICF), and Expectation Maximization (EM) approaches

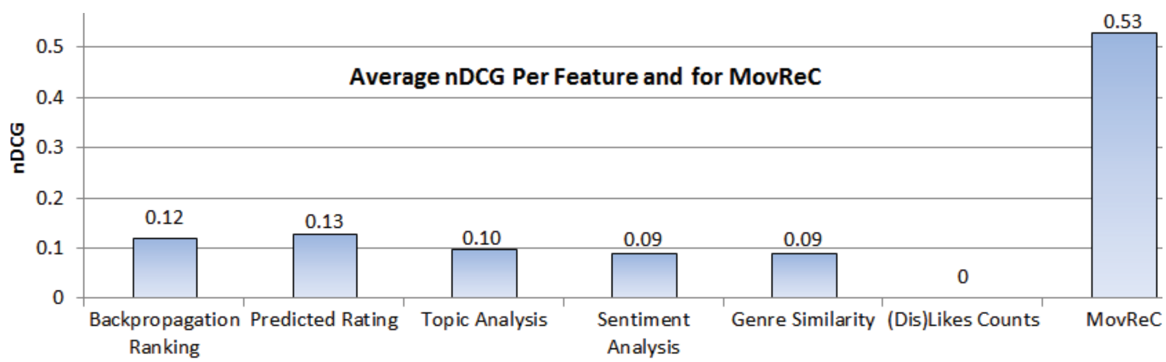


Figure 12: Performance evaluation of MucReC using each individual feature and the combined features

movie suggestions made by MovReC, 53% were parents or guidance of children. Given that (i) parents and guidance are the ones who often select movies for children and (ii) the impossibility of directly interacting with children using Mechanical Turk, it is appropriate to quantify the performance of MovReC reflected by the opinions of parents/guidance of children separately from other appraisers with diverse occupations/professions. The *accuracy* ratios computed using the collected MT appraisers' responses, which reflect the proportion of recommended movies treated as *useful* by independent appraisers among the top-3 suggestions included in each HIT, are shown in Figure 13. As shown in Figure 13, the accuracy ratio calculated according to parents/guidance's responses yield a statistically significant improvement ($p < 0.01$) over the one based on all the collected responses. The fact that appraisers who are parents/guidance appreciate the recommendations made by MovReC more than the general appraisers provides further evidence of the usefulness of MovReC in making personalized children movie recommendations.

4.4.6 Movie Recommendation Systems to be Compared with MovReC

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

- **MF**. Yu et al. [34] and Singh et al. [27] predict ratings on books and movies based on matrix factorization (MF). Yu et al. introduce nonparametric matrix factorization approaches which make use of

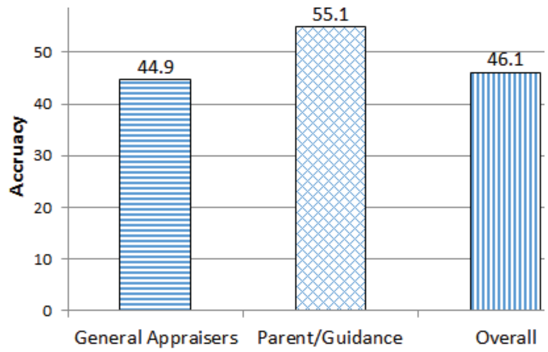


Figure 13: The performance evaluation of MovReC based on the responses of parents/guidance and other appraisers

the singular value decomposition (SVD) and probabilistic principal component analysis (pPCA), two low-rank matrix factorization methods, on large dataset size. Empirical study conducted to verify the perform on the model proposed by Yu et al. demonstrate that nonparametric models outperform other parametric models by making more accurate predictions of user ratings while at the same time are computationally comparable or even faster in training than previous state-of-the-art parametric matrix factorization models. The experimental results verify that nonparametric models are in fact very practical on very large-scale data containing hundreds of millions of ratings.

Singh et al., on the other hand, develop a matrix factorization model, which works efficiently on large, sparse data sets with relational schemas. As shown in [27], relations can encode users' ratings of movies, movies' genres, and actors' roles in movies. Singh et al. have demonstrated that in domains with multiple relations represented as multiple matrices, they can improve predictive accuracy by considering information from one relation while predicting another, i.e., by integrating information from multiple relations, their matrix factorization model can yield better rating predictions. They also verify that it is practical to apply the proposed matrix factorization model on relational domains with hundreds of thousands of entities.

- **ML**. Besides the matrix factorization methods, probabilistic frameworks have been introduced for rating predictions. Shi et al. [26] propose a movie similarity measure based on the movie mood, which is exploited by a joint matrix factorization model for making recommendations. The recommendation model considers not only the user-item rating matrix but also makes use of the contextual information, which is the movie mood, as a regularization term so that the model can learn from user-item matrix and simultaneously allow contextual information to be blended into the recommendation process. Unlike MovReC which relies on online movie features to make recommendations, the joint matrix factorization model is based on the similarity measure to capture reflect the relationship between items with respect to movie mood. Shi et al. suggest movies according to the mood-specific movie similarity in a joint matrix factorization model to improve the context-aware (mood-specific) movie recommendation.

- **Netflix**. We also compare MovReC against the 20 systems that participated in the Netflix contest in 2008. The Netflix Prize solicited for the best collaborative filtering algorithm to predict user ratings for movies given previous ratings on movies, and the grant price of one million dollars was awarded to the team with the lowest Root Mean Squared Error (RMSE) score in predicting user ratings on films based on previous ratings. 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).

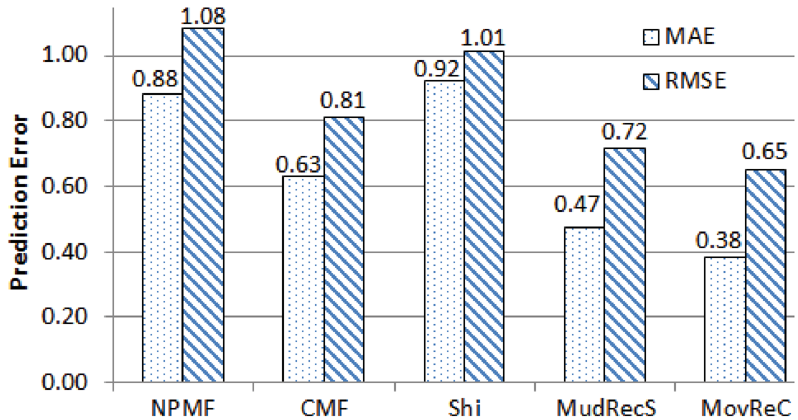


Figure 14: The MAE and RMSE scores for various recommendation systems based on the MovieLens dataset

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

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(f(x_i) - y_i)^2}{n}}, MAE = \frac{1}{n}\sum_{i=1}^{n}|f(x_i) - y_i| \tag{18}$$

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.

- MudRecS [20], 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. MudRecS predicts the *ratings* of multimedia items that match the interests of a user to make recommendations.

Figure 14 shows the MAE and RMSE scores of MovReC and other recommendation systems on the MovieLens dataset. As the MAE and RMSE scores indicate, MovReC significantly outperforms other movie recommendation systems on rating predictions of movies based on the Wilcoxon Signed-Ranks Test ($p \leq 0.05$).

5 Conclusions

Watching movies is one of the popular entertainments in the modern society, and these days people can watch movies anytime and everywhere—at work, at home, or in their cars. However, following the normal supply and demand curve, in the calendar year of 2016 up till mid-July, there were 7,547 most popular English-language movies released.[14] To save time and efforts in searching for children movies appropriate

[14] http://www.imdb.com/search/title?count=100&languages=en &release_date=2016,2016&title_type=feature

and of interest to a user U, for either the educational or entertainment purpose, we propose a personalized recommendation system on children movies called *MovReC*. MovReC is *novel*, since for each candidate movie M to be considered for recommendation to U, MovReC applies (i) the *appropriate ranking analysis* on M to asset its suitability as a children movie, (ii) *rating prediction* on M to determine the likelihood of M preferable by U, (iii) *topic analysis* to find out the theme of M that match U's interests, (iv) *sentiment analysis* and *(dis)like* counts to find out the popularity of M, and (v) *genre analysis* to match the type of movie to which M belongs with the ones previously viewed by U. MovReC is *unique*, since to the best of our knowledge, it is the first personalized children movie recommender. The performance evaluation on MovReC clearly indicates that MovReC makes recommendations that are highly regarded by its users and significantly outperforms other current state-of-the-art movie recommendation systems in terms of prediction accuracy.

References

[1] K. Bacon. Movies, Books, and "The Giver". In *Harvard Educational Magazine*. Harvard Graduate School of Education, Winter edition, January 2015.

[2] D. Blei, A. Ng, and M. Jordan. Latent Dirichlet Allocation. *Machine Learning Research*, 3:993–1022, 2003.

[3] S. Choi, S. Ko, and Y. Han. A Movie Recommendation Algorithm Based on Genre Correlations. *Expert Systems with Applications*, 39(9):8079–8085, 2012.

[4] K. Clarke-Stewart and R. Beck. Maternal Scaffolding and Children's Narrative Retelling of a Movie Story. *Early Childhood Research Quarterly*, 14(3):409–434, 1999.

[5] Q. Diao, M. Qiu, C. Wu, A. Smola, J. Jiang, and C. Wang. Jointly Modeling Aspects, Ratings and Sentiments for Movie Recommendation. In *Proceedings of the* 20^{th} *ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, pages 193–202, 2014.

[6] Film Ratings. *Motion Picture Associate of America Film Ratings*. MPAA.

[7] J. Funk, H. Baldacci, T. Pasold, and J. Baumgardner. Violence Exposure in Real-Life, Video Games, Television, Movies, and the Internet: Is There Desensitization? *Adolescence*, 27(1):23–39, 2004.

[8] T. Griffiths and M. Steyvers. Finding Scientific Topics. *Proceedings of the National Academy of Sciences (PNAS)*, 101(1):5228–5235, 2004.

[9] A. Harding. *Violent Video Games Linked to Child Aggression*. CNN.com, 2008.

[10] F. Harper and J. Konstan. The MovieLens Datasets: History and Context. *ACM Transactions on Interactive Intelligent Systems (TiiS)*, 5(4):Article 19, December 2015.

[11] C. Jacobi, W. van Atteveldt, and K. Welbers. Quantitative Analysis of Large Amounts of Journalistic Texts Using Topic Modelling. *Digital Journalism*, 4(1):89–106, 2016.

[12] K. Järvelin and J. Kekäläinen. Cumulated Gain-Based Evaluation of IR Techniques. *ACM Transactions on Information Systems (TOIS)*, 20(4):422–446, 2002.

[13] H. Jenkins. *Transmedia 202: Further Reflections*. Confessions of an AcaFan. August 1, 2011.

[14] J. Koberstein and Y.-K. Ng. Using Word Clusters to Detect Similar Web Documents. In *Proceedings of the First International Conference on Knowledge Science, Engineering and Management, LNAI 4092*, pages 215–228, 2006.

[15] J. Lee. Analyses of Multiple Evidence Combination. In *Proceedings of the International Conference on Research and Development in Information Retrieval (ACM SIGIR)*, pages 267–276, 1997.

[16] A. Liu, Y. Zhang, and J. Li. Personalized Movie Recommendation. In *Proceedings of the ACM International Conference on Multimedia*, pages 845–848, 2009.

[17] T. Mitchell. *Machine Learning*. McGraw Hill, 1997.

[18] M. Nilashia, O. bin Ibrahima, N. Ithnina, and N. Sarmin. A Multi-Criteria Collaborative Filtering Recommender System for the Tourism Domain Using Expectation Maximization (EM) and PCA-ANFIS. *Electronic Commerce Research and Applications*, 14(6):542–562, October-November 2015.

[19] I. Pilászy and D. Tikk. Recommending New Movies: Even a Few Ratings are More Valuable than Metadata. In *Proceedings of the 3^{rd} ACM Conference on Recommender Systems (ACM RecSys)*, pages 93–100, 2009.

[20] R. Qumsiyeh and Y.-K. Ng. Predicting the Ratings of Multimedia Items for Making Personalized Recommendations. In *Proceedings of the 35th International ACM Conference on Research and Development in Information Retrieval (ACM SIGIR)*, pages 475–484, 2012.

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

[22] M. Rich. *Screen Time Study Finds Education Drop-Off*. The New York Times. January 23, 2014.

[23] L. Rosen. Teaching the iGeneration. *Educational Leadership*, 68(5):10–15, 2011.

[24] N. Sahoo, R. Krishnan, G. Duncan, and J. Callan. Collaborative Filtering with Multi-component Rating for Recommender Systems. In *Proceedings of the 16th Workshop on Information Technologies and Systems*, 2006.

[25] B. Sarwar, G. Karypis, J. Konstan, and J. Reidl. Item-based Collaborative Filtering Recommendation Algorithms. In *Proceedings of the 10th International Conference on World Wide Web*, pages 285–295, 2001.

[26] Y. Shi, M. Larson, and A. Hanjalic. Mining Mood-Specific Movie Similarity with Matrix Factorization for Context-Aware Recommendation. In *Proceedings of the Workshop on Context-Aware Movie Recommendation*, pages 34–40, 2010.

[27] A. Singh and G. Gordon. Relational Learning via Collective Matrix Factorization. In *Proceedings of ACM SIGKDD Conference on Knowledge Discovery and Data Mining (SIGKDD)*, pages 650–658, 2008.

[28] B. Stelter. *8 Hours a Day Spent on Screens, Study Finds*. The New York Times. March 26, 2009.

[29] K. Thompson and F. Yokota. Violence, Sex, and Profanity in Films: Correlation of Movie Ratings with Content. *Medscape General Medicine*, 6(3):Article 3, 2004.

[30] D. Ukpong, U. Uyanga, and I. Nyorere. Cartoon Movies and Nursery Schools Pupils' Development of Cognitive and Social Skills: Implications for Social Studies Teachers. *American Journal of Social Sciences*, 3(3):74–78, 2015.

[31] J. Wakefield. *Children Spend Six Hours or More Day on Screens*. BBC News. March 27, 2015.

[32] J. Yan, N.-Y. Xu, X.-F. Cai, R. Gao, Y. Wang, R. Luo, and F.-H. Hsu. An FPGA-Based Accelerator for LambdaRank in Web Search Engines. *ACM Transactions on Reconfigurable Technology and Systems (TRETS)*, 4(3):Article 25, 2011.

[33] X. Yang, Y. Guo, and Y. Liu. Bayesian-Inference-Based Recommendation in Online Social Networks. *IEEE Transactions on Parallel and Distributed Systems*, 24(4):642–651, April 2013.

[34] K. Yu, S. Zhu, J. Lafferty, and Y. Gong. Fast Nonparametric Matrix Factorization for Large-Scale Collaborative Filtering. In *Proceedings of International ACM Conference on Research and Development in Information Retrieval (SIGIR)*, pages 211–218, 2009.

[35] Y. Zheng, H. Yao, Y. Zhang, and P. Xu. Age Classification Based on Back-Propagation Network. In *Proceedings of the International Conference on Internet Multimedia Computing and Service (ICIMCS)*, pages 319–322, 2013.