Electronic Commerce Research and Applications 21 (2017) 38–49

Contents lists available at ScienceDirect

Electronic Commerce Research and Applications

journal homepage: www.elsevier.com/locate/ecra

# Predicting long-term product ratings based on few early ratings and user base analysis

Sushant S. Khopkar *, Alexander G. Nikolaev

*Department of Industrial and Systems Engineering, State University of New York at Buffalo, Buffalo, NY 14260, United States*

ARTICLE INFO

*Article history:*
Received 2 February 2016
Received in revised form 7 December 2016
Accepted 21 December 2016
Available online 28 December 2016

*Keywords:*
Machine learning
Bayesian network modeling
Product rating prediction

ABSTRACT

Product reviews and ratings are major quality indicators in online shopping systems. In making the decision about carrying a product as part of their inventory, an online seller often pays attention to the product's rating. However, when the observed average is based on a small number of individual user-submitted ratings, the decision-maker may not feel as confident about the product, even when the average is high. The long-term average rating predictions can help online retailers to identify products to promote on their websites as "top picks". The paper proposes a Bayesian Network model to predict the long-term average product ratings based on a (limited) number of early submitted ratings. Performance of the proposed model is compared with the performance of five other prediction methods/models; a Linear Regression model, two variations of a Running Average predictor, an Ordered Logistic Regression model and a Confirmatory Factor Analysis model. Each model's performance is evaluated using the "MovieLens" dataset (GroupLensResearch, 2012). The training is done on 56,590 data points, the ratings submitted for 1155 movies, and the prediction results are reported for 495 movies. It is demonstrated that the proposed Bayesian Network and the Linear Regression models are particularly effective in making accurate predictions around the time of product introduction when the information about the prospective, future ratings is especially valuable. The Bayesian Network model performs better in the very early stage of feedback collection, and in the later stages, as more feedback is received, the Linear Regression model emerges as the best predictor. The strengths and weaknesses of all the assessed prediction methods are discussed.


## 1. Introduction

With the spread and growth of online businesses such as amazon.com and ebay.com, online shopping has become a major form of shopping over the past decade. One of the primary reasons behind the preference for online shopping is that people can buy products without physically visiting stores, which reduces the efforts required for shopping, saves time and allows for quick research about product quality. While shopping online, customers cannot evaluate a product in person, "by the feel". Hence, they often resort to online peer reviews: the textual feedback and reported product ratings people share on retailers' web pages and forum threads. Easy access to reviews and ratings is not limited to products; through this mechanism consumers help each other decide what movies to watch, which restaurants and attractions to visit, and so on. A statistical study (digitalvisitor.com, 2012) revealed that after price considerations, consumers pay the highest attention to the available reviews and ratings in making their buying decisions. Also, 24% of users access online reviews before paying for a service delivered offline (Zhu and Zhang, 2010).

The availability of accurate tools for predicting long-term average product ratings will benefit online retailers to decide which products to promote on their websites as "top picks". It can also help them to forecast demand of a product based on its predicted rating and manage their inventory accordingly. This paper addresses the problem of predicting long-term average ratings of products, with the intent to better inform online retailers of product quality based on early, limited user feedback. In doing so, it relies on available quantitative information about multiple products rated by the retailer's user base. Note that people's buying decisions may be affected not only by numerical product ratings, but also by textual feedback/comments.

These long-term average product ratings can also help consumers in their buying decisions. For any product with its rating information available, two quantities can be observed: (1) the average rating the product has received so far, and (2) the number of peers who have contributed to this rating. If a particular online product has been rated by many users, then the decision-maker is

* Corresponding author.
*E-mail addresses:* skhopkar@buffalo.edu (S.S. Khopkar), anikolae@buffalo.edu (A. G. Nikolaev).

http://dx.doi.org/10.1016/j.elerap.2016.12.002
1567-4223/

likely to trust the information, i.e., rely on it in making her decisions. On the other hand, if a product has been rated by only a few users, then the decision-maker may not feel as confident. For example, when a customer observes an average rating of 4.5 out of 5 for a product while observing that this product has been rated by 47 people, then she feels confident about the quality of the product. She thinks that the "real quality" of the product indeed evaluates to 4.5 out of 5 (on her personal scale, whatever that might be), and hence, she will be comfortable buying this product if its price is in her budget. However, if the same person observes a 5.0 out of 5 rating for another product, while the product has been rated by only two people, then she might believe the rating is subjective and not sufficiently trustworthy (see Fig. 1). Note that the described chain of thoughts can be justified by simple statistics. The more ratings are averaged, the more likely the average is to be close to the true mean of the population of ratings, i.e., the combined opinion of all the potential voters. Hence, long-term average rating prediction is also beneficial to a buyer. The phenomenon of information cascading (Bikhchandani et al., 1992) might also be responsible for this "feeling confident" about a rating when a large of number ratings have already been submitted. However, no such assumption is necessary for the prediction methods considered in this paper. Note that the knowledge of the long-term average product rating may not be as useful to a consumer as it is to a retailer. Indeed, an *average* rating does not capture individual buyer's interests/preferences, while it helps establish an overall interest in the product from all the buyers. As an example, consider the current top rated movie on IBDb website (www.imdb.com), "The Shawshank Redemption", having the average rating of 9.3/10 (rated by 1,734,646 users). This movie comes under the genre of drama. This means that if a person does not prefer the drama genre, then this high average rating might not indicate how she in particular would be impressed by this movie. On the other hand, for a product such as vacuum cleaner, where its performance or dependability is more important than other subjective attributes such as color and design, the determination of long-term average ratings would be more useful to consumers. Meanwhile, in any case, the average product rating is useful to a product distributor to help predict further sales. In summary, the knowledge of the long-term average rating of a product is useful to online retailers, and *may be* useful to consumers.

The challenge of having to use limited information about a product while making judgment of its quality can be addressed by predicting its long-term average rating based on the information about the user base of a specific website. This paper proposes a Bayesian Network model to address the question "Can one predict the long-term average rating for a product (i.e., its rating when a large number of people will have voted on it) that has currently received only a few ratings, using the information of the past ratings given to other products by the same user base?" It also compares the proposed model with five other prediction methods/models.

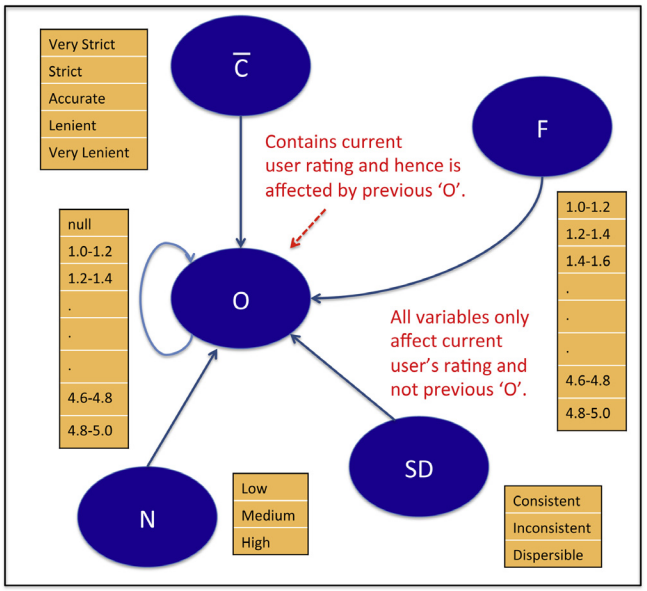



**Fig. 2.** The predictive bayesian network model variables.

## 2. Related work

Few prior research efforts pursued similar objectives. Among those that did is a recent effort of forecasting product adoption that relies on learning the user interaction network characteristics in building projected product diffusion curves based on the observed diffusion trajectories of other products (Trusov et al., 2013). Dover et al. (2012) study the differences in adoption rates of diffusion processes to make inferences about degree distributions of underlying networks, which in turn help to predict product diffusions. The proposed Bayesian Network model in the present paper, however, do not require any knowledge about the users' social network. A prominent research direction relevant to the presented research is recommender system development, which has now become an integral part of many e-commerce sites including amazon.com and ebay.com (Schafer et al., 1999). The relevance comes


**Fig. 1.** The number of ratings combined with the average rating of a product determines users' perspectives towards it.

from the user rating/preference prediction element of the recommender systems. There has been much work done on developing analytical approaches for designing and analyzing such prediction methods (Adomavicius and Tuzhilin, 2005; Resnick and Varian, 1997). The most well studied aspect of these prediction methods is their ability to predict the rating that a specific user might give to a specific product. Conventional prediction logics can be categorized as follows:

- *Content-based prediction logic*: Users' preferences for items are predicted based on the similar items they preferred in the past. These items can be restaurants, city attractions, movies, clothing, electronic products, music albums and so on. Each item has specific characteristics. If a user liked a certain item in the past (or is examining it in the present), items with similar characteristics are predicted for that user. For example, text preference prediction systems like the news filtering system NewsWeeder (Lang, 1995) uses the words of their texts as features. Various types of learning algorithms are used in the content-based prediction methods, such as "Decision Trees and Rule Induction" (Kim et al., 2001; Cohen, 1995), "Nearest Neighbor Methods" (Cohen and Hirsh, 1998; Yang, 1999; Allan, 1998; Billsus et al., 2000), "Relevance Feedback and Rocchios Algorithm" (Rocchio, 1971; Ittner et al., 1995), "Linear Classifiers" (Lewis et al., 1996), "Probabilistic Methods and Naive Bayes" (Maron, 1961; Duda and Hart, 1973). The content-based prediction technique is also called as "item-to-item correlation" (Schafer et al., 1999).
- *Collaborative prediction logic*: Users' preferences for items are predicted based on known preferences of other people with similar tastes and preferences. For example, a music company can analyze user preferences by grouping its users based on what type of music they listen to. Then, using a collaborative filtering algorithm, the company can determine which songs are liked by a group member, so as to predict that those songs would also be liked by the other members of the same group. GroupLens/NetPerceptions (Resnick et al., 1994), Ringo/Firefly (Shardanand and Maes, 1995) and Recommender (Hill et al., 1995) are some of the systems that use this technique. This technique is also called as "people-to-people correlation" (Schafer et al., 1999).
- *Hybrid prediction logic*: Prediction about users' preferences are prepared by using a combination of the collaborative and content-based prediction methods. The simplest type of a hybrid system would be the one which considers weighted linear combination of prediction scores of different systems. The P-Tango system (Claypool et al., 1999) uses such a hybrid. Pazzanis combination hybrid does not use numeric scores, but rather treats the output of each prediction as a set of votes, which are then combined in a consensus scheme (Pazzani, 1999). There are various sub-techniques within hybrid type of prediction logic, such as "Weighted", "Switching", "Mixed", "Feature Combination", "Cascade", "Feature Augmentation", "Meta-level", and so on. Some of the popular algorithms/ applications in these categories are Tran and Cohen (2000), Smyth and Cotter (2000), Basu et al. (1998), Mooney and Roy (2000), Sarwar et al. (1998), Balabanović and Shoham (1997), Condliff et al. (1999), Schwab et al. (2001).

We tackle the problem of predicting a new product's success in a market to determine, e.g., how actively it should be promoted. This problem has not been addressed by the discussed, conventional approaches. Instead of providing product rating predictions for individual users, the objective of this paper is to predict the consensus perceived quality of a product. The proposed approach estimates long-term average ratings, thus providing valuable information about products as they are introduced to a market and while they are rated by only a small number of users. Note that the consensus perceived quality here is understood as the combined opinion of a large number of users which can be quantitatively expressed as the average product rating.

The movie rating database, "MovieLens" (GroupLensResearch, 2012), is used as a testbed for training and testing the model presented in this paper. A movie is a type of an online product in today's world, with companies such as Netflix offering DVDs and online streaming service to their customers. Movies can be rated by customers quantitatively in the same manner as any other online product. Some work on predicting movie ratings/box office revenues has been done in the past (Armstrong and Yoon, 2008; Augustine and Pathak, 2008; Dellarocas et al., 2007; Kim et al., 2015; Neelamegham and Chintagunta, 1999; Zhang et al., 2009), however, unlike in the presented approach, the reported predictions were based on such product-specific factors as names/popularity status of movie directors and actors, movies marketing budget, theatrical availability, distribution strategy and professional critic reviews. (Yu et al., 2012) model sentiments of actual movie reviews and show that both the sentiments expressed in the reviews and the quality of the reviews have a significant impact on the future sales performance. Another manuscript involving working with actual reviews rather than ratings is due to Decker and Trusov (2010). Liu (2006) studied the impact of Yahoo Movies pre-release message board discussions on box office returns and found that the volume of online word of mouth information observed correlates with revenues. Duan et al. (2008) reach a similar conclusion. Chintagunta et al. (2010), however, found that it is not the volume but the valence (mean user rating) that has the real predictive power. These above mentioned studies were focused on predicting successes of movies specifically, and were not applicable to other online products, in general. On the other hand, the methodology proposed in this paper is applicable to any type of online product, where user ratings are available. Also, the objective of the earlier studies was to assess the impact of online word of mouth (WOM) and user ratings on off-line purchase behavior of movies, whereas in this paper, along with comparing the impact of various distinct predictors/factors, we are interested in building an accurate predictive model.

The rest of the paper is organized in the following manner. Section 2 describes the proposed Bayesian Network model and explains how the model parameter learning is conducted using the available data. It also describes a Linear Regression model, two variations of a Running Average predictor, an Ordered Logistic Regression model and a Confirmatory Factor Analysis model as alternative approaches to the problem. Section 3 explains the testing phase, which compares the accuracy of the five prediction methods with the proposed Bayesian Network model over multiple movies in reference to the true long-term movie ratings. Section 4 concludes the paper and discusses future research directions.

## 3. Methods for predicting long-term average product ratings

The prediction methods considered for long-term rating prediction are based on the following assumptions:

(a) the rating a user assigns to a product is influenced by the average product rating she observes at the time of making her assessment (but not by the order of individual votes that produced this average);
(b) the user's rating is also influenced by the observed number of people who have contributed to the product rating she observes at the time of making her assessment;

(c) the user's rating is also influenced by her profile type: this type reflects the user's tendency to generally rate products leniently, accurately or overly strictly;
(d) the user's rating is also influenced by the true quality of the product, as perceived by her;
(e) the user's rating is also influences by the standard deviation of the product ratings she observes at the time of making her assessment (Sun, 2012);

The presented Bayesian Network model and other five learning models are designed, based on these assumptions. In order to present and explain the models' variables in a context, it will be convenient to first describe a specific problem environment. Even though the approach presented in this paper can be applied in a variety of domains, its focus application is media retail, and more specifically, online movie distribution.

### 3.1. Data

The "MovieLens" dataset (GroupLensResearch, 2012) is used for training the models, and also, testing the models' predictions against the true values. The raw training data consists of 80,000 records of movie ratings, with 943 users and 1650 movies. Randomly selected 495 movies (30%) have been removed from the dataset to be used for evaluation purposes. Thus, the resulting training dataset has 56,590 records of movie ratings for 1155 movies, given by 943 users. Note that the raw "MovieLens" dataset was originally composed to include only those users who have rated at least 20 movies. However, working only with significantly contributing users is not a requirement for the model this paper presents: new users or those who have rated just a few movies can be assumed to have the average "leniency" when the actual number of user-rated movies appears insufficient for reliable "leniency" predictions. The raw data are given in the format: "user ID – item ID – rating – timestamp". Here, the "user ID" and "item ID" are the unique identification numbers assigned to each user and each movie, respectively. A "rating" is a user's assessment of the movie. Due to the availability of timestamps, the data is longitudinal, and hence, one can use it to answer questions such as "how many users have rated a given movie before another user has done so?" and "what average rating a given user observed at the time of casting her vote?"

### 3.2. Bayesian network model

The proposed Bayesian Network model is defined using the following variables: the Observed Average Rating ($O$), Number of Available Ratings ($N$), Average Category of Rated Users ($\overline{C}$), Long-Term Average Rating ($F$) and Standard Deviation of the Ratings ($SD$).

*Observed Average Rating (O)*: this is the average rating of a product observed by a user at the time she rates the product. This value can be calculated for each user, for each movie, by using the feedback time stamp information. Since "MovieLens" users assign ratings on the discrete scale of 1–5, $O$ can take any real value between 1 and 5. For training and analysis purposes, a total of 21 states are defined for $O$, dividing its range into intervals of width 0.2. Hence, if the observed average rating for a product is 3.43, $O$ will take the value of "3.4–3.6". This variable has an extra "null" state. Since the first user leaving her feedback on a movie does not observe any running average rating, $O$ assumes the "null" value under this circumstance. Fig. 3 shows the distribution of observed average ratings across the whole training set.

*Number of Available Ratings (N)*: This is the number of people who have already voted for a particular movie at any given fixed timepoint. The value of $N$ is observed from the dataset by using the timestamps of the records. Three states are defined for this variable. When the number of ratings is between 0 and 10, the variable is set to "Low" state. When the number is between 11 and 30, it is set to "Medium" state. If its value is greater than 30, then $N$ is set to "High" state. Table 1 reports the proportion of users who have observed $N$ in the Low, Medium or High states, respectively, while assigning their ratings. Note that the number of states and the range of each state of this variable can vary for different types of products and it should be designed considering the desired level of accuracy. In this paper the range of 'Low' state is kept smaller than that of 'Medium' state, which in turn is kept smaller than 'High' state. The smaller range in the initial period indicates higher precision, where the prediction accuracy is most important for this problem.

*Average Category of Contributing Users ($\overline{C}$)*: All the 943 users contributing to the movie data are categorized into five groups (Very Lenient, Lenient, Accurate, Strict and Very Strict) according to a score based on their overall rating accuracy. This score is the mean of the deviations of their ratings from the final average ratings over all the movies they have rated (in the training dataset). More specifically, let $m$ be the total number of movies user $i$ rated, $R_j^i$ be user $i$'s rating of movie $j$, and $F_j$ be the final average rating of movie $j$, then the score, $S_i$, for user $i$ is calculated as

$$S_i = \frac{1}{m}\sum_{j=1}^{m}(F_j - R_j^i). \quad (1)$$

The value of $S_i$ defines the category of user $i$. Each state of this variable corresponds to a particular range of the category score. Table 2 shows ranges of the category scores and corresponding categories.

Fig. 4 highlights the value of categorizing users into different groups. The bars of the same color correspond to the ratings given by the same user group (see the legends). The values on the $X$-axis in each figure are the final average ratings over all the movies in the training dataset. Observe that very strict users almost always give the rating of 1 to the movies with final average between 2 and 3. Very strict users also often give rating of 1 or 2 to movies with final averages between 3 and 4. On the other hand, observe that when the final average rating of a movie is between 3 and 4, most of the very lenient users give the rating of 5. Similarly, when the final average rating is between 4 and 5, most of the very lenient users give the rating of 5. The distribution of users over different categories resembles a normal distribution (see Fig. 5). The average of the category score of all the users that rated a movie before rating of the user who observes average rating prior submitting his own rating, gives the average category score, and the corresponding state of this variable is the Average Category of Contributing Users ($\overline{C}$).

*Long-Term Average Rating (F)*: The long-term average rating $F$ of any movie falls into the interval [1, 5]. Variable $F$ has 20 states, with its range divided into intervals of width 0.2. Fig. 6 reports the percentages of the movies found in the training dataset for each possible value of variable $F$.

*Standard Deviation of the Ratings (SD)*: It is the standard deviation of the product ratings observed by a user at the time she rates the product. It is calculated using the time stamp information. Three states are considered for this variable: 'consistent', 'inconsistent' and 'dispersed'. Defining the states/levels is up to an analyst; we employed the following procedure. With the 56,590 training data points available and five unique rating gradations (1,2,3,4,5) considered, we calculated two thresholds between the three levels by performing the sampling from two uniform distributions as $\sim$ Uniform(3,4,5) and $\sim$Uniform(1,2,3,4,5). The number of sample

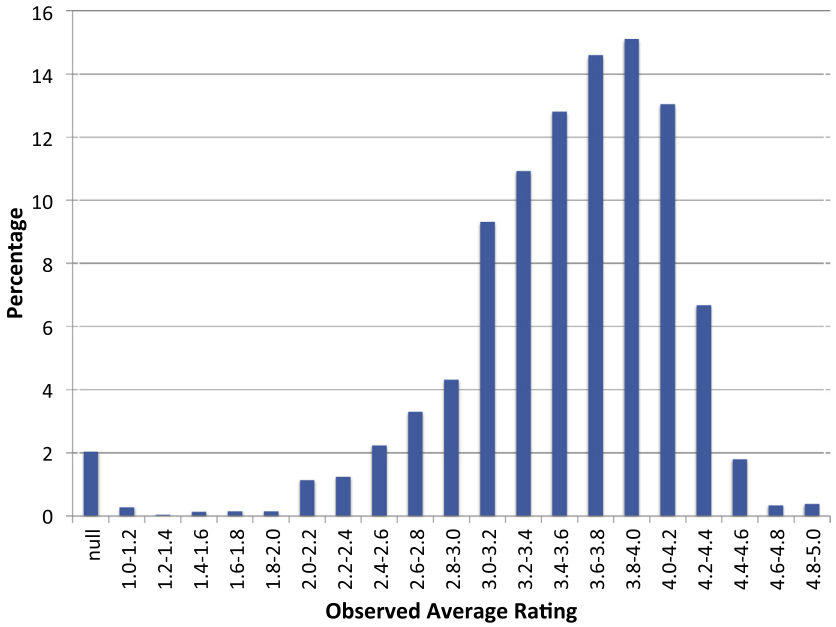



**Fig. 3.** The observed average rating distribution in the training data.

**Table 1**
The distribution of the number of ratings over contributing users.

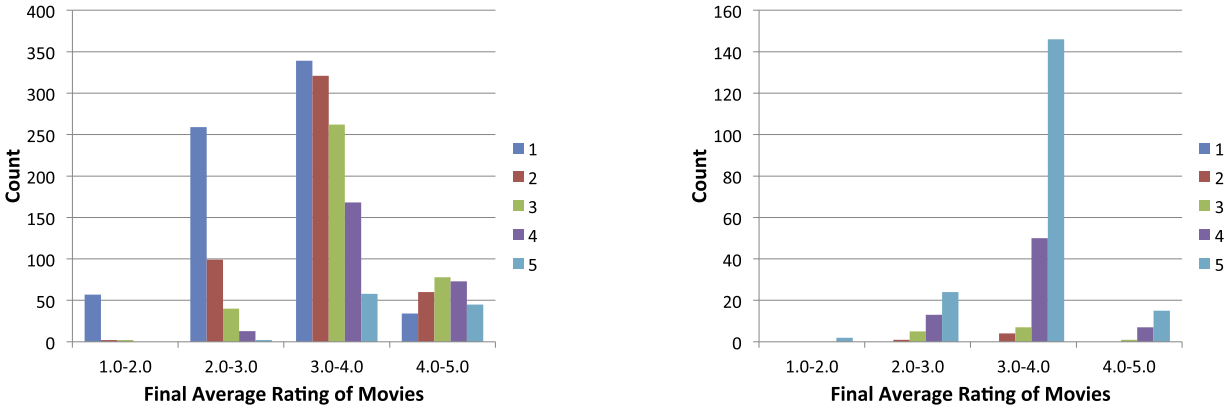
| N | Range | Proportion |
|---|---|---|
| Low | 0–10 | 0.16 |
| Medium | 11–30 | 0.22 |
| High | 31–above | 0.62 |

**Table 2**
Category score ranges and corresponding categories.

| Score range | Category |
|---|---|
| Less than −0.79 | Very Strict |
| Between −0.79 and −0.17 | Strict |
| Between −0.17 and 0.46 | Accurate |
| Between 0.46 and 1.08 | Lenient |
| Greater than 1.08 | Very Lenient |

points is equal to the number of training data points available. Table 3 reports the proportion of users who have observed *SD* in the Consistent, Inconsistent or Dispersed states, respectively, while assigning their ratings.

Variables $O$, $F$, $N$, $\overline{C}$ and $SD$ directly affect the rating of the user, who observes the values of these variables, which is reflected in the observed average rating of the next user. The term "Long-Term Average Rating" is understood as the observed average over all the submitted individual ratings (final average) evaluated after a large number of ratings have been submitted, irrespective of rating submission sequence and times. The long-term average rating of a product can be viewed as consisting of two components: (1) the rating values that every user would provide if "pressed" to evaluate the movie and leave the feedback, and (2) every user's self-selection as a feedback provider (i.e., for every user, the probability that they would rate the product.) Because of the latter, a dependency exists between $F$ and $O$, which we view as being causal and directed from $F$ to $O$; this relationship, along with those for the other model variables, is next discussed in greater detail (see Fig. 2). The number of ratings submitted (so far, with respect to the current user), N, is not the same as the final number of users who would be willing to provide feedback on a particular movie.




(a) Very Strict (b) Very Lenient

**Fig. 4.** The distribution of ratings in the training data for two extreme user categories (very strict and very lenient).

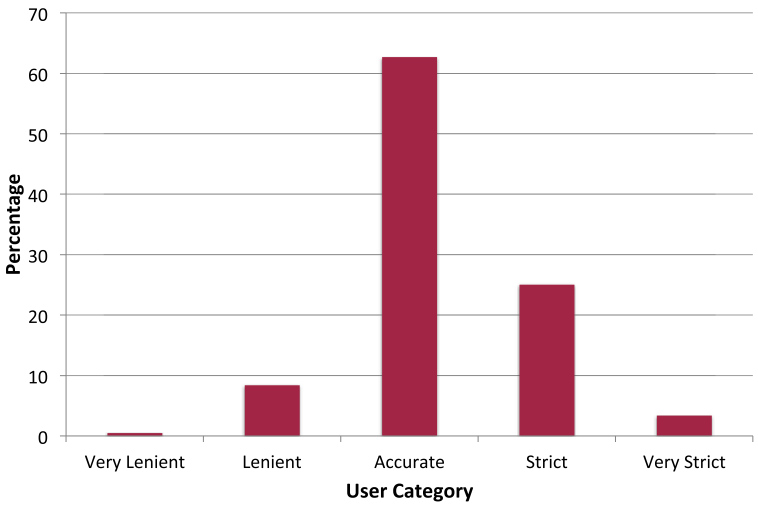



**Fig. 5.** The user category distribution in the training data.

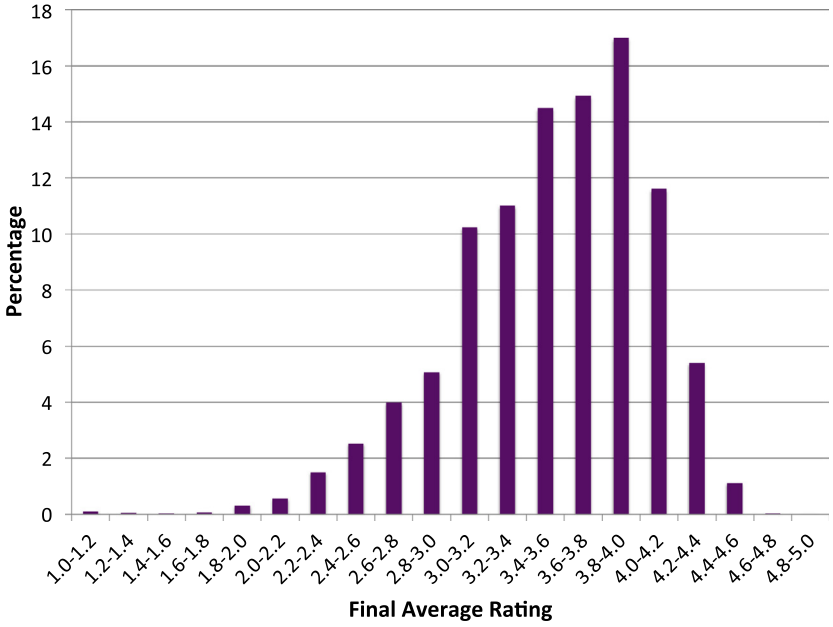



**Fig. 6.** The long-term average rating distribution in the training data.

**Table 3**
The distribution of the standard deviation over all the contributing users.

| N | Range | Proportion |
|---|---|---|
| Consistent | below 0.8 | 0.181 |
| Inconsistent | 0.8–1.4 | 0.796 |
| Dispersed | above 1.4 | 0.233 |

We are not using the latter. Hence $F$ and $N$ are independent. $F$and $\overline{C}$, indeed, may be dependent; however, the Bayesian network we construct, does not rule out such dependence: Nodes $F$ and $\overline{C}$ are parents of $O$; given $O$, they are dependent; this is an indirect consequence of Bayesian Network modeling.

### 3.3. Conditional probability distributions of the model variables

A Bayesian Network is a graph for a model that uses factorization to express a joint distribution of random variables; each variable is represented by a node in this network. A directed arc from node $A$ to node $B$ indicates that $B$ is conditioned on $A$ in the factorization. In a Bayesian Network model, each variable is independent of its non-descendants in the graph given the states of their parents.

Since the described Bayesian Network model structure implies no natural dependencies between variables $N, \overline{C}, SD$ and $F$ (See Fig. 2), the joint probability mass function (*pmf*) $P(N = n, \overline{C} = c, SD = sd, F = f)$ is given as

$$P(N, \overline{C}, SD, F) = P(N) \times P(\overline{C}) \times P(SD) \times P(F).$$

Since node $O$ in the Movie Rating model has four parent nodes, $N, \overline{C}, SD$ and $F$, then one has

$$P(O, N, \overline{C}, SD, F) = P(O|N, \overline{C}, SD, F) \times P(N) \times P(\overline{C}) \times P(SD) \times P(F). \quad (2)$$

The goal of learning is to find the values of the Bayesian Network parameters that maximize the (log) likelihood of the training

dataset. Each probability or conditional probability of a variable in a Bayesian Network is called its parameter. The goal of learning is to estimate these conditional probabilities or parameter values, such that the probability of observing given data is maximized. For example, if node $B$ is the only parent of node $A$ in a Bayesian Network, then the conditional probability $P(A|B)$ is estimated such that the probability of realized values of $A$ and $B$ is maximized. This estimation procedure/ Bayesian Network learning is explained in detail in Section 4.1. The *pmf*s for the variables on the right hand side of Eq. (2) are learnt (estimated) from the training data by counting the number of distinct variable value combinations and dividing this count by the total number of observations. Some values from the joint range of the model variables (those of low likelihood) do not appear in the dataset. In that case, Dirichlet prior (Heckerman et al., 1995; Singer, 1999) is used to estimate the unknown probabilities corresponding to the missing data points. More specifically, suppose that random variable $X$ has *pmf* $P^*$ and can take on values $s \in S, |S| = M$. The training data available for the problem at hand can be viewed as the records of $N$ independent realizations, $x_1, \ldots, x_N$, of $X$. The multinomial estimation problem lies in finding a good approximation of $P^*$ based on the observed training set. This problem can be stated as that of predicting the outcome $x_{N+1}$ given $x_1, \ldots, x_n$. Given a prior distribution over the possible multinomial distributions, the Bayesian estimate of the conditional *pmf* of $x_{N+1}$ is given as

$$P(x_{N+1}|x_1, \ldots, x_N, \xi) = \int P(x_{N+1}|\theta, \xi) P(\theta|x_1, \ldots, x_N, \xi)\, d\theta,$$

where $\theta = \langle \theta_1, \ldots, \theta_M \rangle$ is a vector that describes possible values of the (unknown) probabilities $P^*(1), \ldots, P^*(M)$, and $\xi$ is the "context" variable that denotes additional assumptions about the domain. We assume that data in the training dataset follows Dirichlet prior distribution to estimate the posterior probability distribution for $P^*$. For Dirichlet distribution, the resulting posterior distribution is found in the same distribution family as the prior. More specifically, if the prior is a Dirichlet prior with hyperparameters $\alpha_1, \ldots, \alpha_M$, then the posterior is a Dirichlet distribution with hyperparameters $\alpha_1 + N_1, \ldots, \alpha_M + N_M$. Hence, the conditional *pmf* of $x_{N+1}$ can be expressed as

$$P(x_{N+1} = i|x_1, \ldots, x_N, \xi) = \frac{\alpha_i + N_i}{\sum_j (\alpha_j + N_j)}.$$

Hyperparameter $\alpha_i, i = 1, 2, \ldots, M$ is the number of "imaginary" examples in which one observes outcome $i$. Its value is generally assigned using domain knowledge. The total weight of the hyperparameters represents one's confidence in the prior knowledge.

In the context of the Bayesian Network model, if a particular value combination of $N, \overline{C}, SD$ and $F$ is not observed (available) in the training dataset, the value of the corresponding hyperparameter $\alpha$ is set to 1 (hence, the joint probability of that "imaginary" data point becomes $\frac{1}{1+T}$, if no other data points were added, with $T$ denoting the total number of observations in the training dataset). This means that the model will not dismiss the possibility of observing the same value combination of variables $N, \overline{C}, SD$ and $F$ in the future.

#### 3.3.1. The dynamic nature of the movie rating process

Importantly, the designed Bayesian Network model is time dependent: i.e., it is a Dynamic Bayesian Network (DBN). In a DBN, one or more model variable values in a given period, $t$, affect the corresponding model variable values in the next period, $t + 1$. Also, the structure of the dependency between DBN variables for any two consecutive time periods is the same, which allows for effective and efficient learning. Fig. 7 showcases the dynamic nature of the presented movie rating model. Note that variable $F$ is present in all time periods (though it is not shown in time period 't').

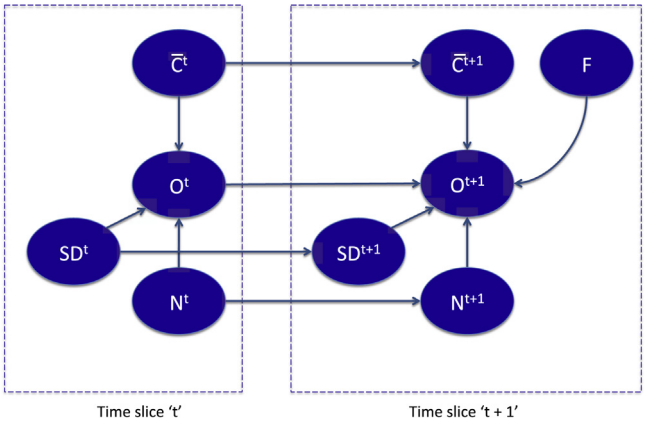



**Fig. 7.** The dynamic bayesian network: a variable dependency graph.

### 3.4. Linear Regression model

The main idea of this work is to design an effective long-term rating prediction tool. Linear Regression, a conventional prediction method in many scientific domains could be a viable alternative to the presented Bayesian Network model. The same training data can be used to parametrize a Linear Regression model; moreover, instead of using categorical values as in the Bayesian Network model, one can use continuous variables in Linear Regression, as follows.

1. Observed Average Rating ($x_1$): $x_1$ is a continuous variable, and hence, is different from variable $O$ defined in the Bayesian Network model.
2. Number of People Already Rated ($x_2$): $x_2$ can take on any positive integer value. It is not categorized, and hence, is different from variable $N$ defined in the Movie Rating model.
3. Category Proportion Score ($x_3$): $x_3$ takes into account the quantitative score assigned to each user category: Very Lenient (-2), Lenient (-1), Accurate (0), Strict (1), Very Strict (2). It is the average of user category values over all the users that have voted on a movie at a point when a new user submits her rating. For example, consider a situation where a movie has been rated by three users. Assume that the categories of these users are Accurate, Strict and Very Lenient, respectively. When the first user rates the movie, the value of $x_3$ is 0. When the second user rates the movie, the value of $x_3$ becomes 0.5 (the average of 0 and 1). Similarly, when the third user rates the movie, the category proportion score (value of $x_3$) becomes $-0.3333$ (average of 0, 1 and $-2$). This negative score indicates that the movie in consideration has been rated somewhat leniently. The influence of this score on the long-term average rating (parameter) is then expressed by the respective regression model estimated from the training dataset.
4. Standard Deviation of Ratings ($x_4$): $x_4$ is a continuous variable, and hence, is different from variable $SD$ defined in the Bayesian Network model.

In addition to the linear terms, second-order terms are also included into the considered Linear Regression model, namely, the quadratic effects $(x_1^2, x_2^2, x_3^2, x_4^2)$ and interaction effects $(x_1 \times x_2, x_1 \times x_3, x_1 \times x_4, x_2 \times x_3, x_2 \times x_4$ and $x_3 \times x_4)$. The regression model parameters are estimated from the same training data with

56,590 entries as the Bayesian Network model. For the 495 movies selected for testing, the model was executed to make rating predictions for the long-term average ratings. Hence, in this model, the long-term average rating is the dependent variable.

### 3.5. Running average predictor

Another quantity commonly perceived as a legitimate indicator of long-term average rating is a running average, i.e., the average over all the observed ratings for a particular movie at a given point in time. Hence, it is expected that as more and more number of ratings become available, the accuracy of this predictor should go on increasing.

### 3.6. A variation of running average predictor

A variation of the Running Average predictor is also considered as a benchmark. We assume a Dirichlet prior distribution, $Dir(\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5)$, of ratings for each movie. Five $\alpha$ parameters are corresponding to the five possible (integer) rating values. The prior distribution of ratings is calculated as the proportion of each rating value encountered among all those in the training dataset (e.g., the prior probability for value 1 ($\alpha_1$) is taken as the count of rating 1 submissions found in the training dataset divided by the sum of all rating counts.

At the prediction stage, this prior distribution is updated to a posterior distribution after each additional rating becomes visible. With the use of the prior, the running average predictor is likely to return better predictions early, i.e., when only a small number of submitted ratings are available. In other words, this predictor is likely to overcome the main limitation of the conventional running average predictor.

### 3.7. Ordered Logistic Regression model

The long-term average ratings of the movies can be categorized into different intervals that are ordinal in nature: the states "1–2", "2–3", "3–4" and "4–5". These states can be rank ordered, but the real distance between these states is not really known. For example, the tendency of moving from the long-term average rating value between 2 and 3 to a rating value between 3 and 4 depends upon the individual user inclinations. As such, a user's inclination of giving rating from 2 vs. 3 may not be the same as choosing rating 4 vs. 5. If this is the case, then an Ordered Logistic Regression model might turn out to be effective. In the presented Ordered Logistic Regression model, the dependent variable is the long-term average rating of a movie. It is categorized into four ordinal groups – "1–2", "2–3", "3–4" and "4–5". The Observed Average Rating ($x_1$, continuous), the Number of People Already Rated ($x_2$, integer), the Category Proportion Score ($x_3$, continuous) and the Standard Deviation of Available Ratings ($x_4$, continuous) are used as the independent variables (same as in the Linear Regression model above).

### 3.8. Confirmatory Factor Analysis model

Fig. 2 implies that the long-term average rating is a latent (i.e., hidden) variable. Hence, a latent measurement model can be used to predict it. Different types of latent variable models are typically recognized according to whether the observable and latent variables are categorical or continuous (see Table 4).

In our case, as the latent variable and all but one observable variables are continuous in nature, *factor analysis* is the most suitable choice of the measurement model. Confirmatory factor analysis (CFA) is a special form of factor analysis and is used to test whether the data fit a hypothesized measurement model. As the structure of our model was already selected (as shown in Figure 2), we employ the confirmatory factor analysis (as opposed to general factor analysis) method. The long-term average rating is treated as the latent variable, whereas the Observed Average Rating ($x_1$, continuous), the Number of People Already Rated ($x_2$, integer), the Category Proportion Score ($x_3$, continuous) and the Standard Deviation of Available Ratings ($x_4$, continuous) are observable variables.

**Table 4**
Selection of a latent measurement model.

| Latent variable | Observable variable | |
|---|---|---|
| | Continuous | Categorical |
| Continuous | *Factor analysis* | *Latent trait analysis* |
| Categorical | *Latent profile analysis* | *Latent class analysis* |

## 4. Testing

In order to test the effectiveness of the proposed Bayesian Network model and other selected prediction methods, 495 movies (30% dataset) were selected from the "MovieLens" dataset. For each movie, at every point in time that a user submits her rating, six model predictions were obtained for the long-term average rating: by the Bayesian Network model, the Linear Regression model, the third by the Running Average predictor, the fourth by the variation of Running Average predictor, the fifth by the Ordered Logistic Regression and the sixth by the Confirmatory Factor Analysis model. The models' performance was compared in reference to the true, observed, long-term average rating for each movie.

### 4.1. Prediction based on the Bayesian Network model

The long-term average rating ($F$) is a parent node in the structure of the Bayesian Network model (See Fig. 2). In the test dataset, values of other variables ($O$, $N$, $\overline{C}$ and $SD$) are either realized or can be calculated, while variable $F$ is latent, or hidden. In order to obtain the value of $F$ for a given combination of $O, N, \overline{C}$ and $SD$, a likelihood maximization method can be used.

The algorithm for predicting the long-term average rating by the Bayesian Network model is based on the following logic. When the first user is about to rate a new movie, she does not observe any prior ratings of this movie: the state of $N$ is "Low". The first user's category is available (the user base is kept constant, so when a user arrives to rate a movie, her category is known), hence the value of $\overline{C}$ is known. As the first user does not observe any average rating, the state of $O$ is "null" and $SD = 0$ or in "Consistent" state. For this combination of observable variables ($N$ = "Low", $O$ = null, $SD$ = "Consistent" and the realized value of $\overline{C}$), a value of $F$ can be identified, amongst 21 feasible states, such that the probability in Eq. 2 is maximized. The information about the conditional probability distribution of $F$ is obtained from the training data. The mean over the range of the obtained state of $F$ is treated as the model's prediction. For example, if the state 3.4–3.6 ensures the maximal likelihood of the observed variable value combination, then 3.5 is taken as the predicted value of $F$. As a more complex example, consider the case where not one but 15 users have contributed their ratings for a given movie. Then, $N$ = "Medium" (as $10 < N < 30$), and the value of $O$ is the state corresponding to the average rating of the first 14 users; $\overline{C}$ and $SD$ are calculated similarly. At this point, the ratings contributed by all the users, from the first to the $15^{th}$, are known: each such rating value will henceforth be referred to as an "observation". Using Eq. (2), the value of $F$ is identified such

that the likelihood of *multiple* observations (15 observations in the given example) is maximized,

$$\max_F P(F, C, O, N, SD) = \prod_{t=1}^{15} P(\overline{C}^t) P(SD^t) P(O^t | O^{t-1}, \overline{C}^t, N^t, SD^t, F) \qquad (3)$$

If two or more states of $F$ result in the same maximal probability of occurrence, then the average of their means is selected as the predicted value. For example, if two such states are 2.4–2.6 and 2.8–3.0, then the average of 2.5 and 2.9 is computed, returning 2.7 as the predicted value of $F$.

### 4.2. Prediction based on the Linear Regression model

Using linear regression, the long-term average rating of a movie is predicted by treating it as variable $y$ that depends on variables $x_1, x_2, x_3, x_4$ and their second-order products. The linear regression is performed using the linear model function 'lm()' in *R*. Refer to Table 5 for regression statistics; all the independent variables except the interaction between $x_2$ and $x_3$ (Number of People Rated and Average Category Score) are found to have significant effect, with $p$-values less than 0.05. The Adjusted R-Square value of 0.8491 confirms that the model has adequate predictive power.

### 4.3. Predictions by the running average predictor and its variation

The running average converges toward, and in fact, *becomes* the long-term average of a movie. The conventional running average predictor is not expected to be robust during the initial stages of predictions, where only a small number of ratings are available, as it does not "learn" in the training stage. Its variation, however, as described in Section 3.6 is expected to overcome this limitation, as it learns the prior distribution of ratings.

### 4.4. Prediction based on the Ordered Logistic Regression model

Using the Ordered Logistic Regression model the categorized variable $y$, of long-term average rating is predicted. $y$ can take four different values, viz., "1–2", "2–3", "3–4" and "4–5". Estimation is done using 'polr()' function in *MASS* package of *R*. The 'predict()' function gives probabilities associated with each state of the dependent variable. Using these probabilities expected value is calculated using the mean value of each state, as the corresponding long-term average rating. For example, if an output of the predict function are probabilities (0.1, 0.3, 0.4, 0.2) for states "1–2", "2–3", "3–4" and "4–5" respectively, then the long-term average rating is predicted as $(1.5 \times 0.1) + (2.5 \times 0.3) + (3.5 \times 0.4) + (4.5 \times 0.2)$. Table 6 show Odds Ratio values and 95% confidence interval (CI).

**Table 5**
Summary of statistics for the predictive linear regression model.

| | Coefficients | $p$-Value |
|---|---|---|
| Intercept | 2.152 | $< 2^{-16}$ |
| $x_1$ | 0.227 | $< 2^{-16}$ |
| $x_2$ | −0.001 | $1.69^{-11}$ |
| $x_3$ | −0.038 | 0.032 |
| $x_4$ | −1.309 | $< 2^{-16}$ |
| $x_1^2$ | 0.029 | $< 2^{-16}$ |
| $x_2^2$ | $-2.302^{-06}$ | $< 2^{-16}$ |
| $x_3^2$ | −0.072 | $< 2^{-16}$ |
| $x_4^2$ | 0.023 | $2.29^{-05}$ |
| $x_1 \times x_2$ | 0.001 | $< 2^{-16}$ |
| $x_1 \times x_3$ | 0.050 | $< 2^{-16}$ |
| $x_1 \times x_4$ | 0.398 | $< 2^{-16}$ |
| $x_2 \times x_3$ | $-6.205^{-05}$ | 0.743 |
| $x_2 \times x_4$ | −0.002 | $< 2^{-16}$ |
| $x_3 \times x_4$ | 0.118 | $< 2^{-16}$ |

**Table 6**
Summary of statistics for the ordered logistic regression model.

| | Odds ratio | 95% confidence interval | |
|---|---|---|---|
| $x_1$ | 756.15 | 683.87 | 837.07 |
| $x_2$ | 1.0047 | 1.0044 | 1.0051 |
| $x_3$ | 8.0682 | 7.0886 | 9.1849 |
| $x_4$ | 0.973919 | 0.8903 | 1.0655 |

If the 95% CI does not cross 0, the parameter estimate is statistically significant. It can be seen from Table 6 all the independent variables are found to be statistically significant.

### 4.5. Prediction based on the Confirmatory Factor Analysis model

The Confirmatory Factor Analysis model treats the long-term average rating ($y$) as a hidden or latent variable and estimates parameter values for the observable variable from the training data. Confirmatory Factor Analysis (CFA) is a subset of the much wider Structural Equation Modeling (SEM) methodology. SEM is provided in *R* via the 'sem' package. Table 7 shows factors and their corresponding estimates. One-headed arrows are assumed to indicate factor loadings or regression coefficients, and two–headed arrows are assumed to indicate variances and covariances. Using the linear combination of reported parameter values, long-term average ratings are predicted.

### 4.6. Comparative analysis of predictor performance

The Mean Square Error (MSE) is a convenient metric that can be used to compare the performance of the proposed Bayesian Network model with the other five selected prediction methods with respect to true long-term averages of the selected 495 movies. By definition, squared error is the square of the difference between a value returned by predictor and the corresponding true value. The average of such squared errors across all predicted ratings for a particular movie gives the MSE of a predictor for that movie. For each movie, the long-term average rating predictions were obtained retrospectively, i.e., at the times that each of the contributing users submitted their ratings. According to Table 8, when

**Table 7**
Summary of statistics for the confirmatory factor analysis model.

| Factor | Parameter estimate | $p$-Value |
|---|---|---|
| $x_1 \leftarrow y$ | 0.9759 | 0.00 |
| $x_1 \leftarrow x_2$ | $-2.5361^{-05}$ | 0.052 |
| $x_1 \leftarrow x_3$ | −0.3059 | 0.00 |
| $x_1 \leftarrow x_4$ | −0.1130 | 0.00 |
| $y \leftrightarrow y$ | 0.2645 | 0.00 |
| $x_1 \leftrightarrow x_1$ | 0.0647 | 0.00 |
| $x_2 \leftrightarrow x_2$ | $4.7383^{03}$ | 0.00 |
| $x_3 \leftrightarrow x_3$ | 0.0405 | 0.00 |
| $x_4 \leftrightarrow x_4$ | 0.0781 | 0.00 |

**Table 8**
Performance comparison of the six prediction methods.

| Method | Mean MSE | | | | |
|---|---|---|---|---|---|
| | Overall Score | First 5 | First 10 | First 15 | First 20 |
| BN | 0.1983 | 0.2299 | 0.1613 | 0.1264 | 0.1024 |
| LR | 0.1817 | 0.2330 | 0.1604 | 0.1209 | 0.0977 |
| RA | 0.1586 | 0.4024 | 0.2539 | 0.1863 | 0.1470 |
| Var RA | 1.157 | 0.9576 | 0.7473 | 0.5877 | .4778 |
| OLR | 0.1976 | 0.3463 | 0.2238 | 0.1655 | 0.1339 |
| CFA | 0.2607 | 0.5759 | 0.3513 | 0.2508 | 0.1962 |

BN: Bayesian Network; LR: Linear Regression; RA: Running Average Var; RA: Variation of Running Average; OLR: Ordered Logistic Regression; CFA: Confirmatory Factor Analysis.

the average MSE across all the 495 test movies is compared for the six prediction methods, the Running Average predictor (Mean MSE = 0.1586) performed better than all the other methods. The Bayesian Network model (Mean MSE = 0.1983), the Linear Regression model (Mean MSE = 0.1817) and the Ordered Logistic Regression model (Mean MSE = 0.1976) have comparable performances. The performance of the Confirmatory Factor Analysis (Mean MSE = 0.2607) is not as good as these methods. The Variation of the Running Average predictor (Mean MSE = 1.157) has the lowest performance. The latter predictor performs well for some of the movies, but does particularly bad on a sizable set of others: these are the movies whose true long-term average ratings are low (and hence, they are rare, as shown in Fig. 6), with relatively few people rating them. The small number of ratings does not give enough time for the method to let the prior distribution turn into an acceptable posterior distribution. Moreover, though the posterior distribution becomes more and more accurate with a greater number of ratings available, unlike the conventional running average predictor, the prediction converges to the true long-term average rating very slowly. Table 9 shows some statistics of the number of ratings per movie, such as mean, variance, minimum and maximum values in the training and test datasets.

Even though the Running Average predictor performed better than the other prediction methods, having been averaged over all the contributed ratings, it is more important to compare the predictors' performance at the time that only the first few ratings were submitted for each particular movie. The third, fourth, fifth and sixth columns of Table 8 shows the performance of the prediction methods when only first 5, 10, 15 and 20 ratings are available respectively. It can observed that the proposed Bayesian Network model and the Linear Regression model performed far better than the Running Average predictor in the initial period. Fig. 8 demonstrates it nicely. Note that the curves belonging to the Bayesian Network model and the Linear Regression model are almost completely overlapping on each other. Hence, the curve of the Linear Regression model is made dotted. It is interesting to note that the predicted variable *F* in the Linear Regression model is continuous in nature, while in case of the Bayesian Network model it is categorical (20 states of *F* are considered). It means it might be possible to improve performance of the proposed Bayesian Network model further by considering a higher number of states of all the variables, provided that large training data is available. The Running Average predictor is particularly off the mark at the initial rating stages. This is because at these stages, the Running Average predictor relies heavily on the consensus opinion of a biased subpopulation of users who just happened to watch the movie before everyone else. Indeed, early on, it is hard to predict exactly which subset of users will end up rating a particular movie. This point deserves a more in-depth discussion. All the considered predictors strive to overcome the bias that arises due to self-selection of users in the reviewer pool. In doing so, the proposed Bayesian Network model and the Linear Regression model uses prior information to estimate the review pool composition, namely, it estimates what would happen if the whole population voted on a typical movie. The Running Average predictor does not use any prior information, and simply accepts the submitted ratings.

**Table 9**
Statistics of the number of ratings per movie.

| Dataset | The number of ratings per movie | | | |
|---|---|---|---|---|
| | Mean | Variance | Minimum | Maximum |
| Training | 48.995 | 4275.5 | 1 | 422 |
| Testing | 47.295 | 4111.9 | 1 | 484 |

**Table 10**
Robustness of Bayesian network predictions.

| Method | Mean MSE | | | | |
|---|---|---|---|---|---|
| | Overall Score | First 5 | First 10 | First 15 | First 20 |
| BN (Original) | 0.1983 | 0.2299 | 0.1613 | 0.1264 | 0.1024 |
| Mean of 25 experiments | 0.2186 | 0.2478 | 0.1779 | 0.1354 | 0.1139 |
| SD of 25 experiments | 0.0080 | 0.0130 | 0.0120 | 0.0083 | 0.0061 |

BN: Bayesian Network; SD: (Sample) Standard Deviation.

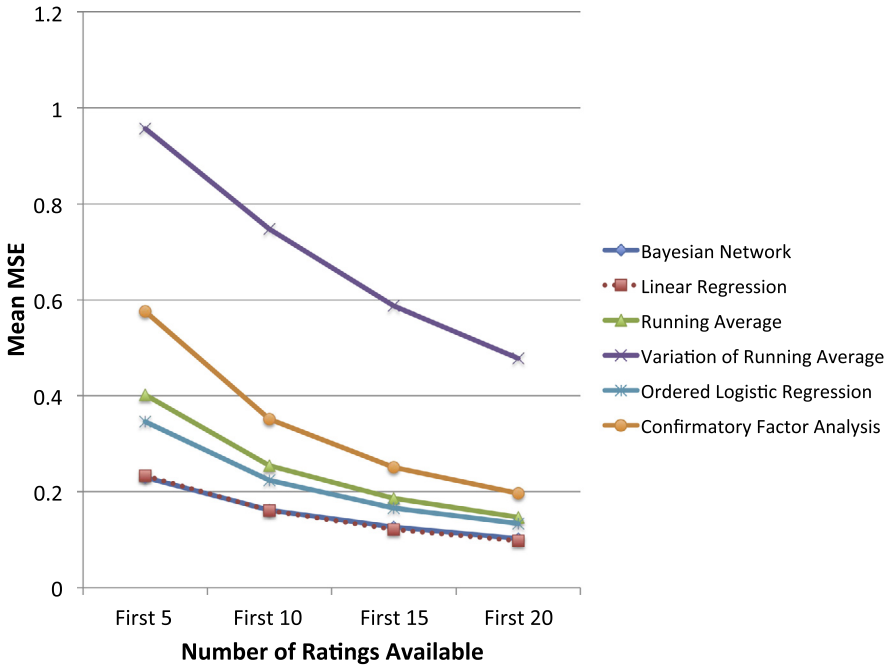



**Fig. 8.** Comparison of the six prediction methods based on mean of MSE.

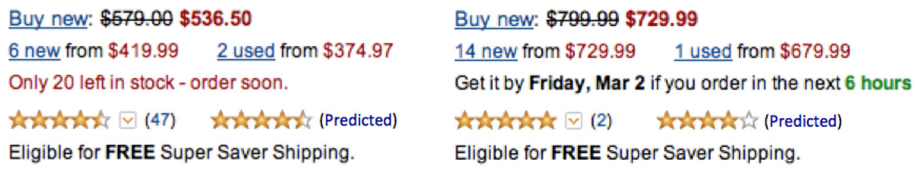



**Fig. 9.** A possible design of the user rating information box implementing the proposed concept.

It is interesting to note that long-term average rating predictions given by the Bayesian Network model are fairly robust in nature and hence the need for outlier detection can be avoided. Outlier detection is necessary when one is interested in making point prediction of a single rating. However here, even though we are doing point predictions, our goal is to predict long-term average ratings. Hence, even if a few extreme ratings are present, their effect is diminished when we consider long-term averages. To demonstrate this, we performed the following experiment:

For all 495 movies in the test data set, we randomly shuffled the sequence in which users appear to rate movies. This experiment is repeated 25 times for each scenario (scenario 1: predicting when first 5 ratings are available; scenario 2: predicting when first 10 ratings are available; and so on). This random shuffling allows "extreme" ratings to appear in any order and in any position. The aim here is to assess an expected effect of outlier rating on the mean square error in predicting long-term average ratings. The results of our investigation are summarize in Table 10. It can be observed that the mean values of Mean MSE over the 25 experiments are very close to the original Mean MSE values for the Bayesian Network. Moreover, the small values of sample standard deviations indicate that the scores are quite robust and would not be affected much by outliers.

Fig. 9 proposes an interface design of a product description incorporating the long-term average rating prediction concept.

## 5. Conclusion and directions for future research

This paper addresses the challenge of designing a tool capable of predicting long-term average product ratings that can help online retailers make better-informed decisions about managing their inventory, potentially leading to increased sale volumes. It may also partially help customers in their buying decisions for products that are purchased based on their objective performance, rather than individual preferences. To do so, the paper proposes a Bayesian Network model to make predictions and considers five other prediction methods to compare prediction performances. The "MovieLens" dataset of movie ratings is used for the training and testing of the models. The relative effectiveness of the proposed Bayesian Network model and the five other prediction methods is demonstrated by comparing their prediction results with the true long-term averages. It is observed that overall, as assessed by the Mean Square Error (MSE) measure, the Running Average predictor performs the best. The performance of the proposed Bayesian Network model, the Linear Regression model and the Ordered Logistic Regression model is slightly inferior; the performance of the Confirmatory Factor Analysis is average; the Variation of Running Average predictor performed the worst among all the methods.

The proposed Bayesian Network model and the Linear Regression model performs better than the other prediction methods, especially when the running average information is most limited, and hence, when the prediction accuracy is most valuable. The Bayesian Network model performs better in the very early stage of feedback collection, and in the later stages, as more feedback is received, the Linear Regression model emerges as the best predictor. The Running Average predictor is effective when large number of ratings are available, and can be very sensitive when small number of ratings are available. Performance of the Bayesian Network model might be further improved by considering more number states associated with its variables, provided large amount training data is available. Outlier impact analysis shows that long-term average rating predictions given by the Bayesian Network model are quite robust in nature and hence the need for outlier detection can be avoided. If sentiment analysis is performed on textual feedback available, a quantitative score representing this feedback can be considered as an additional factor in the prediction models.

Online retailers can readily implement the approaches mentioned in the paper. The amount of training data will have the highest impact on prediction accuracy. Hence, one can imagine that the big companies such as amazon and ebay would have an upper edge over small scale online retailers. The dataset used for the models' testing relied upon rich information about the user base. Future research can explore the effectiveness and robustness of the predictors across various datasets having different amounts of information available for training the model.

One might be interested in designing a hybrid model that would combine the advantages of the two model based predictors (Bayesian Network and Linear Regression) and the Running Average predictor. Such a hybrid model could dynamically decide which one of the three predictors should be displayed as the best option for predicting the long-term average of a product, as a function of the number of available user ratings ($N$). The hybrid model itself can be imagined as a new Bayesian Network model, where predictions by the three predictors and the number of available ratings ($N$) would be independent variables (parent nodes), while the predictor selector (selecting one model/predictor over the other) would be a multinomial dependent variable (child node).

Thus, the presented research not only demonstrates a useful real-world application of prediction models, but it also creates a few interesting research directions.

## Acknowledgement

The authors would like to thank Mr. Harshalkumar Patel for his valuable help in computational work. Alexander Nikolaev was partially funded by the Academy of Finland, Grant 268078 (MineSocMed).

## References

Adomavicius, G., Tuzhilin, A., 2005. Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. IEEE Trans. Knowledge Data Eng. 17 (6), 734–749.

Allan, J., 1998. Topic detection and tracking pilot study: final report. In: Proc. DARPA Broadcast News Transcription and Understanding Workshop.

Armstrong, N., Yoon, K., 2008. Movie rating prediction. Tech. rep. Carnegie Mellon University.

Augustine, A., Pathak, M., 2008. User rating prediction for movies. Tech. rep. Citeseer.

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

Basu, C., Hirsh, H., Cohen, W. et al., 1998. Recommendation as classification: Using social and content-based information in recommendation. In: AAAI/IAAI. pp. 714–720.

Bikhchandani, S., Hirshleifer, D., Welch, I., 1992. A theory of fads, fashion, custom, and cultural change as informational cascades. J. Polit. Economy, 992–1026.
Billsus, D., Pazzani, M.J., Chen, J., 2000. A learning agent for wireless news access. In: Proceedings of the 5th International Conference on Intelligent User Interfaces. ACM, pp. 33–36.
Chintagunta, P.K., Gopinath, S., Venkataraman, S., 2010. The effects of online user reviews on movie box office performance: accounting for sequential rollout and aggregation across local markets. Market. Sci. 29 (5), 944–957.
Claypool, M., Gokhale, A., Miranda, T., Murnikov, P., Netes, D., Sartin, M., 1999. Proceedings of ACM SIGIR workshop on recommender systems, vol. 60. Citeseer.
Cohen, W.W., 1995. Fast effective rule induction. ICML, vol. 95, pp. 115–123.
Cohen, W.W., Hirsh, H., 1998. Joins that generalize: Text classification using whirl. KDD, pp. 169–173.
Condliff, M.K., Lewis, D.D., Madigan, D., Posse, C., 1999. Bayesian mixed-effects models for recommender systems. Proc. ACM SIGIR, vol. 99. Citeseer.
Decker, R., Trusov, M., 2010. Estimating aggregate consumer preferences from online product reviews. Int. J. Res. Market. 27 (4), 293–307.
Dellarocas, C., Zhang, X.M., Awad, N.F., 2007. Exploring the value of online product reviews in forecasting sales: The case of motion pictures. J. Interact. Market. 21 (4), 23–45.
digitalvisitor.com, 2012. The importance of online customer reviews. Accessed: 05/04/2012. URLhttp://www.digitalvisitor.com/latestnewsandresources/social-media-blog/the-importance-of-online-customer-reviews.html
Dover, Y., Goldenberg, J., Shapira, D., 2012. Network traces on penetration: uncovering degree distribution from adoption data. Market. Sci. 31 (4), 689–712.
Duan, W., Gu, B., Whinston, A.B., 2008. The dynamics of online word-of-mouth and product sales–an empirical investigation of the movie industry. J. Retail. 84 (2), 233–242.
Duda, R.O., Hart, P.E., et al., 1973. Pattern Classification and Scene Analysis, Vol. 3. Wiley, New York.
GroupLensResearch, 2012. Movielens data sets. Accessed: 04/25/2012. URLhttp://www.grouplens.org/node/73
Heckerman, D., Geiger, D., Chickering, D., 1995. Learning bayesian networks: the combination of knowledge and statistical data. Mach. Learn. 20 (3), 197–243.
Hill, W., Stead, L., Rosenstein, M., Furnas, G., 1995. Recommending and evaluating choices in a virtual community of use. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. ACM Press/Addison-Wesley Publishing Co., pp. 194–201.
Ittner, D.J., Lewis, D.D., Ahn, D.D., 1995. Text categorization of low quality images. In: Symposium on Document Analysis and Information Retrieval. Citeseer, pp. 301–315.
Kim, J.W., Lee, B.H., Shaw, M.J., Chang, H.-L., Nelson, M., 2001. Application of decision-tree induction techniques to personalized advertisements on internet storefronts. Int. J. Electron. Commerce 5, 45–62.
Kim, T., Hong, J., Kang, P., 2015. Box office forecasting using machine learning algorithms based on SNS data. Int. J. Forecast. 31 (2), 364–390.
Lang, K., 1995. Newsweeder: Learning to filter netnews. In: Proceedings of the Twelfth International Conference on Machine Learning. Citeseer.
Lewis, D.D., Schapire, R.E., Callan, J.P., Papka, R., 1996. Training algorithms for linear text classifiers. In: Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval. ACM, pp. 298–306.
Liu, Y., 2006. Word of mouth for movies: its dynamics and impact on box office revenue. J. Market. 70 (3), 74–89.
Maron, M.E., 1961. Automatic indexing: an experimental inquiry. J. ACM (JACM) 8 (3), 404–417.
Mooney, R.J., Roy, L., 2000. Content-based book recommending using learning for text categorization. In: Proceedings of the fifth ACM conference on Digital libraries. ACM, pp. 195–204.
Neelamegham, R., Chintagunta, P., 1999. A bayesian model to forecast new product performance in domestic and international markets. Market. Sci. 18 (2), 115–136.
Pazzani, M.J., 1999. A framework for collaborative, content-based and demographic filtering. Artif. Intell. Rev. 13 (5–6), 393–408.
Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., Riedl, J., 1994. Grouplens: an open architecture for collaborative filtering of netnews. In: Proceedings of the 1994 ACM conference on Computer supported cooperative work. ACM, pp. 175–186.
Resnick, P., Varian, H., 1997. Recommender systems. Commun. ACM 40 (3), 56–58.
Rocchio, J., 1971. Relevance feedback in information retrieval. In: SMART Retrieval System Experiments in Automatic Document Processing.
Sarwar, B.M., Konstan, J.A., Borchers, A., Herlocker, J., Miller, B., Riedl, J., 1998. Using filtering agents to improve prediction quality in the grouplens research collaborative filtering system. In: Proceedings of the 1998 ACM conference on Computer supported cooperative work. ACM, pp. 345–354.
Schafer, J.B., Konstan, J., Riedi, J., 1999. Recommender systems in e-commerce. In: Proceedings of the 1st ACM conference on Electronic commerce. ACM, pp. 158–166.
Schwab, I., Kobsa, A., Koychev, I., 2001. Learning user interests through positive examples using content analysis and collaborative filtering. Internal Memo, GMD, St. Augustin, Germany.
Shardanand, U., Maes, P., 1995. Social information filtering: algorithms for automating word of mouth. In: Proceedings of the SIGCHI conference on Human factors in computing systems. ACM Press/Addison-Wesley Publishing Co., pp. 210–217.
Singer, N., 1999. Efficient bayesian parameter estimation in large discrete domains. Adv. Neural Inf. Process. Syst. 11 (11), 417.
Smyth, B., Cotter, P., 2000. A personalised tv listings service for the digital TV age. Knowledge-Based Syst. 13 (2), 53–59.
Sun, M., 2012. How does the variance of product ratings matter? Manage. Sci. 58 (4), 696–707.
Tran, T., Cohen, R., 2000. Hybrid recommender systems for electronic commerce. In: Proc. Knowledge-Based Electronic Markets, Papers from the AAAI Workshop, Technical Report WS-00-04. AAAI Press.
Trusov, M., Rand, W., Joshi, Y.V., 2013. Improving prelaunch diffusion forecasts: using synthetic networks as simulated priors. J. Market. Res. 50 (6), 675–690.
Yang, Y., 1999. An evaluation of statistical approaches to text categorization. Inf. Retrieval 1 (1–2), 69–90.
Yu, X., Liu, Y., Huang, X., An, A., 2012. Mining online reviews for predicting sales performance: a case study in the movie domain. IEEE Trans. Knowledge Data Eng. 24 (4), 720–734.
Zhang, L., Luo, J., Yang, S., 2009. Forecasting box office revenue of movies with bp neural network. Expert Syst. Appl. 36 (3), 6580–6587.
Zhu, F., Zhang, X., 2010. Impact of online consumer reviews on sales: the moderating role of product and consumer characteristics. J. Market. 74 (2), 133–148.