Journal of Retailing 96 (3, 2020) 328–343

Automated Product Recommendations with Preference-Based Explanations☆

André Marchand[a,*], Paul Marx[b]

[a] Chair of Marketing, Leipzig University, Grimmaische Straße 12, 04109 Leipzig, Germany
[b] Department of Marketing, University of Siegen, Hoelderlinstr. 3, 57068 Siegen, Germany

Available online 20 January 2020

Abstract

Many online retailers, such as Amazon, use automated product recommender systems to encourage customer loyalty and cross-sell products. Despite significant improvements to the predictive accuracy of contemporary recommender system algorithms, they remain prone to errors. Erroneous recommendations pose potential threats to online retailers in particular, because they diminish customers' trust in, acceptance of, satisfaction with, and loyalty to a recommender system. Explanations of the reasoning that lead to recommendations might mitigate these negative effects. That is, a recommendation algorithm ideally would provide both accurate recommendations and explanations of the reasoning for those recommendations. This article proposes a novel method to balance these concurrent objectives. The application of this method, using a combination of content-based and collaborative filtering, to two real-world data sets with more than 100 million product ratings reveals that the proposed method outperforms established recommender approaches in terms of predictive accuracy (more than five percent better than the Netflix Prize winner algorithm according to normalized root mean squared error) and its ability to provide actionable explanations, which is also an ethical requirement of artificial intelligence systems.

Keywords: Recommender systems; Recommendation explanations; Decision support systems; Consumer preferences; Netflix; MoviePilot

Introduction

Intensive uses of smart devices are moving society in general, and online retailing in particular, into an age of artificial intelligence–based assistance provided by automated recommender systems (RS). For online retailing, these RS both support and influence consumer choices, in that they seek to reduce consumer choice complexity by analyzing past behaviors and then generating personalized recommendations (Bodapati 2008; Hennig-Thurau, Marchand and Marx 2012; Zhang et al. 2019). In doing so, RS reduce consumers' search costs while increasing the retailer's up- and cross-selling potential and competitive advantages (Chung, Rust, and Wedel 2009; Syam and Kumar 2006). Accordingly, RS have grown common, prompting personalized results on online retail sites, search engines, social networks, and news, music, and video sites (Lee and Hosanagar 2019). Their popularity likely stems from consumers' preferences to avoid being overloaded by too many, irrelevant offers. Consumers prefer to view only those items that might meet their needs, ideally with minimal effort on their part, while still maintaining choice autonomy (André et al. 2018; Fleder and Hosanagar 2009).

Although prior research has significantly improved recommender accuracy by developing and enhancing numeric algorithms (Ricci et al. 2015), the algorithms remain subject to potential errors, such as those resulting from insufficient or incomplete data, algorithmic processing errors, or misspecification of the decision model (Aksoy et al. 2006; Herlocker, Konstan and Riedl 2000). If retailers like Amazon in turn issue erroneous recommendations to customers, the credibility of the RS and the firm suffer, potentially even leading to customer rejection, which would diminish the firm's reputation and the customer's satisfaction (Fitzsimons and Lehmann 2004; Gershoff, Mukherjee, and Mukhopadhyay 2003). This fundamental issue limits the practical acceptance and utility of RS for

☆ The authors are listed as per alphabetical order

* Corresponding author.
E-mail addresses: mail@andre-marchand.de (A. Marchand), paul.marx@uni-siegen.de (P. Marx).

https://doi.org/10.1016/j.jretai.2020.01.001
0022-4359/

retailers and consumers. However, we note evidence that explanations for a recommendation can mitigate the risks of erroneous recommendations (Bleier and Eisenbeiss 2015; Tintarev and Masthoff 2007). Such explanations thus might increase trust and better meet ethical requirements for future artificial intelligence systems (Vincent 2019).

We propose incorporating an ability to provide explanations into the base framework of a recommendation algorithm. Improving recommendation algorithms and generating useful explanations represent complementary considerations in this sense, though prior research only addresses them separately. Therefore, we present a recommendation method that addresses both questions concurrently. Rather than attempting to construct an explanation facility around pre-calculated recommendations, the proposed algorithm can design explanations that are meaningful and understandable to each individual user. In essence, our proposed recommendation method is a hybrid of a classic item-based collaborative filtering process and a novel attribute-based preference elicitation technique. The latter technique extracts attribute preferences from consumers' past ratings and generates recommendations based on attribute part-worths. The peculiarity of this task is that insufficient data points (ratings) generally are available to estimate individual customer part-worths. Our technique overcomes this issue with a multi-stage estimation procedure that first obtains interval parameter estimates using auxiliary univariate regressions, then corrects these estimates for omitted variable bias, and finally optimizes the parameter values within confidence intervals to achieve reliable estimates of individual part-worths.

In the next section, we present a parsimonious overview of relevant prior research and detail the interplay of recommender algorithms and explanation styles, to substantiate the need for a novel recommendation method. We develop our modeling framework and present the novel attribute elicitation technique, then test it empirically using two large, real-world data sets. The proposed method outperforms other state-of-the-art recommendation algorithms (including hybrid ones) in both predictive accuracy and the ability to provide actionable explanations. This observation aligns with our suggestion that attribute-based preference models cannot capture product preferences for some users—a problem that our switching hybrid method addresses well. We conclude with a discussion of our findings and directions for further research.

Literature Overview

Recommendation Algorithms

Techniques for estimating personalized ratings consist of three broad categories: collaborative filtering, content-based filtering, and hybrid approaches (Ricci et al. 2015). *Collaborative filtering* (*CF*) exploits information about the preferences of an entire user base to produce recommendations. The set of CF methods encompasses three types that vary in how they use rating data: *user-based* CF, *item-based* CF, and *matrix factorization* (MF).

In user-based CF approaches, users who have exhibited preferences similar to the preferences of the current user in the past provide predictions of the current user's future preferences. Aggregated ratings from similar users for a particular item anticipate the rating that the current user would offer for that item (Konstan et al. 1997; Ricci et al. 2015). Item-based CF examines item profiles to find those that are similar to items that the target user liked in the past, then predicts the ratings by aggregating the user's favorite item ratings, weighted by their similarity to the item in question (Linden, Smith and York 2003; Sarwar et al. 2001). Matrix factorization approaches are a little different from user-based and item-based CF, in that they do not examine similarities between users or items. Instead, they employ rating data from all users to derive a set of latent factors that describe hidden associations between users and items. These approaches characterize both users and items in a multidimensional joint factor space, using factors automatically inferred from the ratings (Koren, Bell, and Volinsky 2009). Conceptually, they mimic principal component analysis, though they modify the procedure for applications to scarce data sets in which the majority of data points are missing (Jolliffe 2014).

Content-based (CB) approaches instead rely on the attribute preferences of a target user to find items similar to those that the user has preferred in the past. The calculation of item similarity in CB methods relies not on the ratings of other users but rather on item characteristics (Pazzani and Billsus 2007), such as counts of distinct words in text documents or the presence or level of a specific product attribute. These recommendation techniques all have merits and limitations that require trade-offs with respect to which approach any particular RS should employ. Table 1 summarizes their various strengths and weaknesses according to the criteria for a good recommendation system, which we discuss next.

The first criterion for a good RS is that it still works when only a few users have rated the same items. This challenge refers to the sparsity of the underlying database (Burke 2002). Consider Amazon, a firm that stocks millions of items for sale to millions of users. It would be unrealistic to anticipate that any individual user could rate a considerable percentage of the items in Amazon's catalog; rather, each customer might rate only vanishingly small subsets of all the items offered by this firm. Thus, CF approaches cannot fulfill this criterion. For MF approaches, though the sparsity problem has not been sufficiently investigated, there are some indications that it might be mitigated somewhat, because these methods reduce the dimensionality of the space for the recommendations, by extracting latent factors from the original data (Burke 2002). Finally, CB approaches do not rely on ratings from other users for their predictions. Rather, they use item content descriptions, available for each item in the catalog, as the basis for their recommendations. Thus, the item space of a CB recommender is dense, and the density of its user space is irrelevant, suggesting that CB approaches may be less likely to suffer from critical concerns about sparsity.

Next, cold start problems, which can be designated *new user* and *new item*, pertain to users or items for which the RS lacks sufficient information to be able to generate an accurate rating prediction (Konstan et al. 1997). This challenge is especially

Table 1
Criteria for good recommendation approaches.

Approach Criteria	User-based collaborative filtering (CF)	Item-based collaborative filtering (CF)	Matrix factorization (MF)	Content-based (CB)	Proposed approach
Works if few users have rated the same items	×	×	×	√	√
Works for new users	×	√	×	√	×
Works for new items	√	×	×	√	√
Prevents uniformity	√	√	√	×	√
Works for users with unusual tastes	×	√	√	√	√
Works for less popular items	√	×	√	√	√
Is not prone to malicious attacks	×	×	×	√	√
Works with preference changes	×	×	×	×	√

acute for user-based approaches, because the RS needs knowledge about active users, gleaned from their ratings, to identify users with similar preferences (Jannach et al. 2011). Similarly, new items must receive ratings before they can be recommended by an RS. This early rater issue arises because users who provide the first ratings for new items receive little benefit (Avery and Zeckhauser 1997). As a subclass of CF approaches, MF approaches suffer equally from both new user and new item problems. However, they rely less than other RS approaches on the similarity between users or items; instead, MF processes involve factorizing the matrix entries, in a manner that is largely independent of the row or column affiliations of these entries. Cold start problems are not problematic for CB approaches.

A good RS should prevent uniformity, frequently referred to as a portfolio effect or overspecialization (Burke 2002; Jannach et al. 2011; Linden, Smith and York 2003). Because CB approaches recommend items that match an active user's profile, they tend to generate recommendations for items that are very similar to those that these users have already seen or purchased (Adomavicius and Tuzhilin 2005). This criterion is no problem for CF and MF approaches. Moreover, a good RS should work for users with unusual tastes. The "gray sheep" problem exists because users with unusual tastes are hard to assign to a neighborhood of similar users; their unusual rating profiles do not correlate well with the rating profiles of other users (Claypool et al. 1999). This problem is prominent for user-based CF. In the related problem of starvation, less popular items might be starved, while popular items become easier to find as more users rate them, which is a problem for item-based CF. The starvation effect thus can decrease sales diversity (Lee and Hosanagar 2019).

A good RS also should not be susceptible to *malicious attacks* (i.e., shilling), in which unethical actors inject ratings into the system explicitly to lower or raise the popularity of a particular item (Ricci et al. 2015). If a vendor wants to increase its revenues earned from an independent online store, it might compromise the retailer's RS by creating several user profiles, issuing ratings that conform with the preferences of its target customers, and then using these profiles to assign high ratings to the vendor's own products and low rankings to competitors' products. This problem arises for CF and MF but not for CB approaches.

Finally, a good RS should be able to accommodate preference changes, rather than become overly rigid or insensitive to changes in user preferences. Established knowledge about the prior preferences of a user can easily come to dominate any new user input. Imagine that a devoted science fiction fan abruptly begins assigning high ratings to dramatic films. An RS might not recognize these changes in the user's preferences, particularly if the new input conflicts with some prior negative ratings for dramas, and instead might dismiss the new ratings as outliers and continue to fail to recommend dramatic films. Our proposed approach is designed to fulfill all these criteria. Its main limitation is that it requires new users to rate at least six products.

Hybrid Recommendation Systems

To transcend these trade-offs and challenges, existing hybrid systems generate recommendations by combining CF and CB methods (Burke 2002). They thus compensate for the limitations of either individual recommendation method and incorporate the benefits of both. Jannach et al. (2011) identify three basic hybridization designs: monolithic, pipelined, and parallelized. Monolithic versions integrate multiple approaches into a single recommender component. In pipelined hybrids, multiple approaches build on each other, step by step. For example, the approach that won the Netflix Prize "blended" the predictions of 109 predictor sets produced by different algorithms (Bell et al. 2009). The contribution of each algorithm to the final rating (i.e., weight of each algorithm) is determined by a linear regression, in which the dependent variable is the vector of the ratings in the holdout set, and the independent variables are the vectors of the ratings predicted by different methods for the same training set.

Finally, a parallelized or switching hybrid uses the predictions of one component if the other component fails to produce recommendations with a sufficient level of confidence (Billsus and Pazzani 2000). Both monolithic and pipelined hybrids disconnect preference-relevant item attributes from the recommendation process, thus limiting their ability to provide meaningful explanations. The switching hybrid can explain the reasons for recommendations though. Thus, a RS that strives to produce both accurate recommendations and comprehensive explanations should implement a switching hybrid.

Recommendation Explanations

The ability of RS to provide explanations, with appropriate content and levels of detail, depends on the algorithm used to develop those recommendations (Herlocker, Konstan and Riedl 2000). In a famous attempt to improve these algorithms, the Netflix Prize competition prompted vast investigations into the accuracy of RS algorithms. Netflix granted researchers access to a rating data set with more than 100 million ratings, provided by approximately 500,000 anonymous consumers, with notable implications for RS research; many studies have relied on this data set (Bennet and Lanning 2007). The resulting concentration on rating data has been aggravated by limitations of contemporary information processing algorithms, which cannot automatically extract meaningful product characteristics (Ricci et al. 2015). In turn, various product characteristics have not been addressed in extant recommender research.

Many consumers express limited trust in recommendations, possibly because they fail to offer relevant explanations and provide only vague statements instead (e.g., Amazon's "recommended based on your browsing history"). Furthermore, automated recommendations are inherently prone to errors, because their generation relies on sparse and incomplete data (Herlocker et al. 2004). If consumers recognize an erroneous recommendation and feel motivated to determine the reason for it, they usually confront a black box, such that the RS does not enable them to reconstruct the process by which the recommendations were determined. This lack of transparency, combined with erroneous recommendations, may impair consumers' acceptance of and trust in the RS (Herlocker, Konstan and Riedl 2000).

Explanations instead should increase transparency and provide users with a comprehensible means to deal with errors (Cramer et al. 2008; Herlocker, Konstan and Riedl 2000). Tintarev and Masthoff (2007) expect that explanations increase consumers' acceptance of, trust in, and loyalty to a RS provider. In addition, good explanations enable users to uncover important choice criteria that they might not have regarded as relevant and to resolve preference conflicts by contributing preference-relevant information that makes the optimal option more evident. Explanations also help users form their own judgments about recommendations, assess the suitability of the recommendations for their decision contexts, and evaluate recommendations more efficiently, all of which should increase the quality of users' choices (Chen 2009;), choice efficiency, and satisfaction with the chosen item (Tintarev and Masthoff 2012).

For these benefits of explanations to be realized though, the explanations must be comprehensible. The ability of a RS to provide clear explanations depends on the recommendation technique being employed (see Bilgic and Mooney 2005). For example, user-based CF approaches offer recommendations that refer to user profile similarities in statements that connect a target user to other users who also have rated the recommended item (e.g., "Customers who bought item X also bought items Y and Z"). Reflecting the underlying process, this explanation style is referred to as a *nearest neighbor* style. In contrast, item-based CF recommenders operate according to item similarities and invoke links between the recommended item and items that the target user has bought or rated in the past (e.g., "X is being recommended because you bought items Y and Z"). This explanation style emphasizes how items have influenced user preferences, so it reflects an *influence* style.

In contrast, CB approaches use attribute-level preferences to generate recommendations, so the systems address individual item attributes that are relevant to the formation of consumer preferences and choice. Because item attributes typically are extracted from the content of recommended items and summarized as keywords, they constitute *keyword* styles, as exemplified by phrases such as, "This item received a high relevance score, because it contains the terms f_1, f_2, and f_3" (Bilgic and Mooney 2005; Symeonidis, Nanopoulos and Manolopoulos 2008). Gedikli, Jannach, and Ge (2014) find that CB explanations are particularly helpful for increasing user-perceived transparency and satisfaction with recommendations. As this discussion indicates, the nearest neighbor style is associated with user-based CF, the influence style implies item-based CF, and the keyword style reflects CB approaches. MF techniques do not permit either of these explanation styles; instead, they base their recommendations on uninterpretable factor solutions.

The ability of hybrid systems to provide explanations, using any particular style, depends on the extent to which each hybrid method uses "pure" predictions of the individual components in its final recommendations. In switching hybrids for example, the style of the explanation is dictated by the component whose prediction led to the recommendation. If a hybrid system can attribute its final recommendation to one of its components, it can explain the recommendation. An explanation style in a hybrid system might combine keyword and influence explanation styles to offer, "Item X is recommended because it contains features a and b, which also appeared in items Z and W, which you rated positively." However, hybrids that mix the results of their components, using some aggregation rule, also lose the ability to attribute the resulting recommendation to a specific approach, so they may struggle to explain their recommendations (Ricci et al. 2015).

Finally, prior research has established the effectiveness of various explanation styles with respect to user acceptance and decision quality. Bilgic and Mooney (2005) show experimentally that users who receive nearest neighbor explanations tend to overestimate the quality of the recommended items, which may lead to mistrust and cause them to stop using the RS. They also find that explanations provided with an influence or keyword style are significantly more effective than those relying on the nearest neighbor style. The keyword style slightly dominates the influence style and also provides advantages of convenience and effectiveness; users do not have to engage in as much inference in this case. Symeonidis, Nanopoulos and Manolopoulos (2008) find that explanations that combine keyword and influence styles dominate either individual style. Thus, we conclude that a hybrid recommendation algorithm should combine content- (keyword) and item- (influence) based CF approaches.

Model Framework

We propose explicitly incorporating product characteristics to both increase predictive accuracy and enable the generation of detailed explanations for the recommendations. In particular, capturing product attribute–related preferences may offer more information than rating data, as well as support more flexible derivations of consumer preferences, at a finer level of resolution, during recommendation generation. In turn, this process may lead to more precise predictions of consumers' preferences for particular items.

To introduce our framework, we begin with a simple model and refine it, introducing new variables step by step to capture consumer preferences. Particularly, we differentiate three effects: static effects that result directly from user–item interactions, static effects that can be attributed solely to a user or an item, and temporal effects.

Basic Model

In recommender systems, holistic preferences are represented by ratings that the system acquires from its users, either explicitly or implicitly. We model the rating as the result of the interaction of a user's attribute-based preferences and item attributes. This basic model resembles the widely acknowledged weighted additive decision rule:

$$r_{u,i} = \mu + \sum_{j \in J} m_{i,j} p_{u,j} + \varepsilon_{u,i} \tag{1}$$

where $r_{u,i}$ is the rating that consumer u assigns to product i. The constant term μ denotes the centered baseline for user part-worths. We use the mean value of all ratings contained in a recommender system's database as the value of μ, which statistically represents the expected value of a user's rating in a situation in which no preference information about the user is available in the system. In accordance with the law of large numbers, if high numbers of ratings are examined (a condition frequently fulfilled by an RS), the sample mean will converge to the expected value of the rating. Furthermore, $m_{i,j}$ can be either binary, indicating the presence or absence of a specific nominal product attribute (e.g., brand, color), or metric, which indicates the quantity of a measurable product attribute (e.g., price, weight, age). Finally, $p_{u,j}$ denotes the preference of the consumer for the j th product attribute from the overall set of product attributes $J \ni \{1..n\}$, and $\varepsilon_{u,i}$ is an error term. This model assumes that product ratings are known from users' past rating records and that product characteristics are readily available or can be derived from a particular source, such as a product catalog. The part-worths $p_{u,j}$ must be estimated.

Static Effects Beyond the User–Item Interaction

Eq. (1) models the rating solely as the product of interactions between item attributes and a user's attribute part-worths. However, some effects may be independent of this interaction and associated solely with users or items. According to prior recommender literature, different users apply rating scales in diverse ways. For example, some users systematically assign higher ratings than others (e.g., Adomavicius and Tuzhilin 2005; Sarwar et al. 2001), which causes the mean rating of an individual user to deviate from the overall mean, an effect we refer to as the *user fixed effect*. Similarly, an *item fixed effect* may result from various causes, such as the popular appeal of mainstream products, so that the mean ratings of these products rises higher than the overall mean rating (Jannach et al. 2011; Ricci et al. 2015).

Users also may differ in their reactions to the average ratings and popularity levels of products. One group of users might adapt to mainstream assessments, another group could respond to such assessments overly positively, and a third group might express skepticism and rate products in ways that defy general trends. These reactions involve both users and products, yet they occur at a general level that does not relate to attribute-level interactions; that is, they change the user's rating of a product as a whole, independent of the composition of attributes and their levels in the product profile. Incorporating these effects into our model leads to the following expression:

$$r_{u,i} = \mu + b_u + b_i s_u + \sum_{j \in J} m_{i,j} p_{u,j} + \varepsilon_{u,i} \tag{2}$$

where $b_u = \bar{r}_{u,*} - \mu$ denotes user fixed effects, defined as the deviation of a user's mean rating value from the overall mean rating. Analogously, item deviation is $b_i = \bar{r}_{*,i} - \mu$. We capture the user's reactions to item fixed effects with the scale factor s_u.

Accounting for Time

Because RS rely on historical data, our model must account for temporal effects. We base the model extension on Koren (2009). In the course of time, consumers may change their preferences and rating behaviors; some products become classics, whereas others fall into oblivion. To capture temporal effects, we extend the static parameters of our model with time-varying parameters and replace the term b_u in Eq. (2) with $b_u + \alpha_u t$. In this expression, α_u is the slope of a user's long-term rating trend, and we redefine b_u as the static portion of a user's rating. Analogously, we extend the item fixed effects and user reaction factors to $b_i + \beta_i t$ and $s_u + \gamma_u t$, respectively, where β_i and γ_u denote the slopes of the long-term trends of the item fixed effects and the user's reaction to it. Similarly, the user's part-worths can be reconstructed as $p_{u,j} + \delta_{u,j} t$, which is the slope of the long-term change in a user's preference toward the jth product attribute. These modifications produce our final model:

$$r_{u,i} = \mu + b_u + \alpha_u t + (b_i + \beta_i t)(s_u + \gamma_u t) + \sum_{j \in J} m_{i,j}(p_{u,j} + \delta_{u,j} t) + \varepsilon_{u,i} \tag{3}$$

Estimating Model Parameters

Once estimated, the values of the parameters in Eq. (3) can be used to predict the user's ratings of new and unfamiliar

products and provide explanations for the recommendations. These explanations might reflect positive attributes that had the most influence in leading to a particular recommendation, reveal which negative attributes exerted negative preference effects, or highlight the relative importance of the attributes or the directions of their temporal changes in a user's decision-making process. Yet in many cases, Eq. (3) cannot be solved algebraically, or else its parameters cannot be estimated simultaneously with statistical techniques, such as regression analysis, due to the scarce data that are inherent to big data sets. In particular, when the number of preference-relevant product attributes exceeds the number of ratings provided by a specific user, the model in Eq. (3) remains underdetermined.

Many professionals rely on Bayesian estimation frameworks to estimate underdetermined models. However, with respect to the high data dimensionality of real-world RS, and considering the many parameters that we need to estimate for our model, an iterative Bayesian estimation procedure would require inordinate computer resources and immense computational time. These requirements are overly burdensome for practical applications, for which the model parameters must be updated as soon as new data are available. Therefore, we propose an alternative, two-step procedure to estimate the parameters of the underdetermined model. First, we derive an initial solution to Eq. (3) by running a set of regressions per parameter, with a subsequent correction for the omitted variable bias. Second, we optimize the initial parameter values using a modified gradient descent procedure.

Estimation Step

Data insufficiency hinders the simultaneous estimation of all model parameters at once, so we propose running a set of univariate regressions to obtain the initial parameter estimates for each user and each parameter separately. To do so, we apply ordinary least squares (OLS), which provides inferences about parameter significance and access to confidence limits for parameter values. With these values, we can interpret the OLS results as interval estimates and constrain the optimization routine in the second step of our method. We use parameter significance information to remove those that have no statistical meaning for describing the product preferences of a particular consumer. This information also can indicate the certainty of the system with respect to provided recommendations. However, the individual estimation of parameters introduces an underspecification error to OLS models, which may lower the quality of the obtained individual estimates. Therefore, we examine the consequences of an underspecification of an OLS regression and present a method to counteract them.

Counteracting the Omitted Variable Bias

Machine learning methods and regularization procedures search for an optimum in the trade-off between bias and variance. The resulting estimates then can support predictions. Too much bias or too much variance in estimates both result in poor predictions. Because we initially estimate all parameters in univariate regressions, their values are likely overestimated (due to omitted variable bias; see Gujarati 2009, pp. 471–473). For our estimation procedure, this overestimation becomes a problem for forecasting the rating, so we need to correct the values of the parameters obtained from the univariate auxiliary regressions for the overestimated part that is due to the omission of relevant predictors in the auxiliary regressions. We can counteract the consequences of underspecification to some extent by exploiting the properties of the biased estimates and reducing these risks. This procedure for correcting the omitted variable bias is detailed in the Appendix. It can be generalized inductively for an arbitrary number of parameters of the "true" regression model. We next present the details of the estimation of the model parameters from Eq. (3).

Estimation of User- and Item-Related Effects

As we described in the model framework section, user fixed effects, item fixed effects, and the user's reaction to item popularity should be conceptually independent, both from one another and from user–item interactions, such that they should be unaffected by the omitted variable problem. Thus, we perform bivariate auxiliary regressions to determine the appropriate initial parameter values, their significance, and their confidence intervals. We begin by estimating the user fixed effects parameters b_u and α_u. For each user, we conduct an OLS regression of the form $r_{u,i} = b_u^{'} + \alpha_u t$. We derive the consumer's rating trend parameter α_u directly from this regression, whereas the baseline b_u is recovered from $b_u^{'}$ by subtracting the overall rating mean from $b_u^{'}: b_u = b_u^{'} - \mu$. We choose $p > .10$ as the general cut-off criterion for the regression parameters. With respect to time resolution, we select t to denote the number of days elapsed since the user's first rating.

Because new users require time to become accustomed to a RS, we assume that rating behaviors change more rapidly for new than for experienced users. To prevent overfitting the regression to unstable fluctuations of a user's average rating, the user must provide product ratings for at least 120 days (SD = 60 days),[1] so that we can capture drifts in rating behavior. For consumers who do not meet this condition or who indicate insignificant values of α_u, the parameter gets discarded from the model, and $b_u^{'}$ is calculated as the mean of the consumer's ratings. In these cases, the confidence limits of $b_u \pm \sigma t$ are established for b_u, with t obtained from Student's t-distribution for $p < .10$, and the degrees of freedom are one less than the number of ratings by the consumer. Furthermore, $\sigma = \sqrt{\sum (r_{u,i} - \mu)^2 / df}$ is the standard deviation of differences between a user's ratings and the overall mean. We estimate the item fixed effects in a similar fashion, using auxiliary regressions of the form $r_{u,i} = b_i^{'} + \beta_i t$. However, we expect slower changes in prod-

[1] This number is somewhat arbitrary, but in sensitivity analyses, we tested 120 days for about one-third of a year, 240 days for about two-thirds of a year, and 90 and 150 days with no significant changes.

uct popularity and therefore require the time frame between an item's first and last ratings to be at least 240 days.

The estimates for the parameters that capture a consumer's reaction to deviation then can be determined in two steps. First, we fix the user and item parameters in Eq. (3) at their estimated values and ignore the features of the model that address user–item interactions, such that we set $\sum_{j \in J} m_{i,j} p_{u,j} = 0$. Given the fixed parameter values for each consumer's rating, we can calculate the difference between the actual rating and the user fixed effects, $r'_{u,i,t} = r_{u,i} - \mu - b_u - \alpha_u t$; we also determine the value of the item fixed effects, $b''_{i,t} = b_i + \beta_i t$. Second, we solve the following regression problem:

$$r'_{u,i,t} = s_u + \gamma_u b''_{i,t} \tag{4}$$

The underlying rationale is that γ_u should capture the portion of the rating that is unrelated to user fixed effects but that varies with time and item fixed effects. Because $b''_{i,t}$ accounts for both of these factors and $r'_{u,i,t}$ is representative of user fixed effects, the estimate for γ_u from Eq. (4) provides precisely what we need. The regression's constant term s_u captures the stable portion of this effect. For users who display insignificant values of both regression parameters, we discard γ_u and s_u from the model.

Estimation of Attribute Part-Worths

In contrast with user and item fixed effects, consumer attitudes toward product attributes are not necessarily mutually independent. In determining the model parameters, we inevitably encounter concerns of model underspecification and must account for biases in the parameter estimates and their variances in our auxiliary regressions. Although we have proposed a method to correct for omitted variable bias, we also may confront a problem of multicollinearity, associated with the risk of poor estimation of the coefficients b_{ji} in the auxiliary regressions (specified in Eq. (A4) of Appendix A). However, the joint effect of the two highly correlated variables can be estimated (with Eq. (A3)). We exploit this property to mitigate the problem of multicollinearity for our model and argue that knowledge about the joint effects of two or more highly correlated variables is sufficient to describe consumer preferences in the model represented by Eq. (3). If certain attributes occur (nearly) jointly in products, their individual relative contributions to a consumer's preferences become irrelevant, because these attributes always affect preferences in combination. From each pair of attributes that correlate highly, defined by $r_{ji} \geq .90$, we eliminate the attribute with lower variance, because it is less helpful for discriminating between items. This elimination occurs at the global level, not separately for each examined user.

Next, we perform regressions of the following form to estimate the attribute part-worth parameters of each user:

$$r_{u,i} = \beta'_0 + p_{u,j} m_{i,j} + \delta_{u,j} m_{i,j} t + \varepsilon_{u,i} \tag{5}$$

where $p_{u,j}$ and $\delta_{u,j}$ are the parameters of interest, which designate the static and time-dependent components, respectively, of user $u's$ preference for the j th attribute of product i; β'_0 is the constant term of the regression; and t is the time elapsed, in days, since the first rating appeared in the data set. Similar to the process of user fixed effects estimation, we require a consumer to have rated products for at least 120 days before we attempt to capture the time-varying components of the part-worths. For consumers who do not fulfill this requirement, we discard $\delta_{u,j}$ and estimate the following simplified regression:

$$r_{u,i} = \beta'_0 + p_{u,j} m_{i,j} + \varepsilon_{u,i} \tag{6}$$

This reduced model also can apply when the complete model in Eq. (5) cannot be estimated due to data insufficiency. To correct for omitted variable bias, we ran the set of pairwise auxiliary regressions (specified by Eq. (A4)) and obtained auxiliary parameters b_{ji} that characterize the level of interdependence of the product attributes. The values of the insignificant coefficients b_{ji} are set to 0, because such effects introduce no bias into the underspecified model. This procedure takes place across the whole user–item matrix, rather than at the level of individual users. The auxiliary parameters b_{ji} then can be pooled with the parameters estimated in Eqs. (5) and (6) to create (analogous to Eq. (A5)) a system of J equations:

$$\hat{\alpha}_i = \beta_i + \sum_{j=1}^{J} \beta_j b_{ji} \tag{7}$$

where $\hat{\alpha}_i$ denotes the estimated value of the i th parameter (i.e., $p_{u,j}$ or $\delta_{u,j}$), β_i denotes the unbiased value of this parameter, and $j \in \{1, \ldots, J\}$ designates the index of each remaining parameter. To solve this equation system, we employ singular value decomposition (SVD), which can handle poorly conditioned systems of linear equations in a way that provides an optimal solution, in terms of least squares (Press et al. 2007). Finally, using the solution to Eq. (7), we recalculate the variances of the estimated parameters (in accordance with Eq. (A7)). At this point, we can complete the test for parameter significance. We discard parameters that do not reach significance and estimate the confidence limits for the remaining parameters.

Optimization Step

We next exclude the parameter estimates that describe users' preferences for a product and the set of associated confidence intervals, which we treat as interval estimates of the corresponding parameters. To increase the prediction ability of our preference model (Eq. (3)) while retaining its ability to explain the reasoning for the recommendations in terms of product attributes, we employ a *conjugate gradient method* that iteratively optimizes the initial parameter values by minimizing the loss function associated with our preference model (Press et al. 2007).

Considering the specific nature of our task, we implement three adjustments: initialization of the starting point for the optimization, restriction of the optimization procedure to the confidence limits of the parameters, and measures to prevent model overfitting. That is, we first initialize the optimization process with the parameter values obtained in Step 1. We then restrict the optimization area to the confidence limits of the

parameters determined in the estimation step. Thus, we prevent the optimization procedure from going beyond the scope of possible solutions that likely contain the true parameter values and from reaching a local minimum that might satisfy the restrictions of the loss function but would provide unreliable estimates of users' preferences in terms of the model in Eq. (3). This measure also ensures that the optimized parameter values account for omitted variable bias, because they remain within the interval in which the true values are most likely to occur.

To counteract the threat of overfitting and ensure that the model is generalizable and suitable for predictions of future ratings, we employ a holdout set of six randomly drawn ratings for each user. We exclude the ratings of the holdout set from the entire procedure of estimating and optimizing the parameter values. Instead, we use them in the gradient method to determine a stop point for the optimization that prevents overfitting. In particular, we stop the optimization after an iteration that fails to decrease the loss function for both the learning set and the holdout data.

In accordance with its definition, we calculate the gradient of the loss function as a set of partial derivatives of each parameter of the model. In each iteration, these parameters adjust in a direction opposite the gradient, by a magnitude proportional to the overall step size, as described in Eq. (8). Here, λ_k denotes the step size in the direction of the k th parameter, determined by the conjugate gradient procedure, and $e_{u,i} = r_{u,i} - \hat{r}_{u,i}$ designates the prediction error of a consumer's ratings, calculated on the basis of the parameter values of the current iteration of the optimization process:

$$\begin{aligned}
&1.\ b_u \leftarrow b_u + \lambda_{b_u}\left(-2e_{u,i}\right) \quad 5.\ s_u \leftarrow s_u + \lambda_{s_u}\left(-2e_{u,i}\left(b_i + \beta_i t\right)\right)\\
&2.\ \alpha_u \leftarrow \alpha_u + \lambda_{\alpha_u} t\left(-2e_{u,i}\right) \quad 6.\ \gamma_u \leftarrow \gamma_u + \lambda_{\gamma_u} t\left(-2e_{u,i}\left(b_i + \beta_i t\right)\right)\\
&3.\ b_i \leftarrow b_i + \lambda_{b_i}\left(-2e_{u,i}\left(s_u + \gamma_u t\right)\right) \quad 7.\ p_{u,j} \leftarrow p_{u,j} + \lambda_{p_{u,j}} m_{i,j}\left(-2e_{u,i}\right)\\
&4.\ \beta_u \leftarrow \beta_u + \lambda_{\beta_u} t\left(-2e_{u,i}\left(s_u + \gamma_u t\right)\right) \quad 8.\ \delta_{u,j} \leftarrow \delta_{u,j} + \lambda_{\delta_{u,j}} m_{i,j} t\left(-2e_{u,i}\right)
\end{aligned} \tag{8}$$

Using this procedure, we obtain the final estimates for the parameters of the model of consumer preferences in Eq. (3) and thus can predict consumers' future ratings, as well as provide personalized recommendations. Knowing the values of the model parameters and their significance also enables us to generate explanations of the reasoning for the recommendations.

Hybridization Step

We regard practical applicability as an important constraint on the development of our recommendation method. Therefore, we account for users who do not form product preferences on the basis of product attributes but instead reflect their intrinsic dispositions and base their decisions on factors that extend beyond product characteristics and are difficult to measure, such as anticipated emotions, social pressures, overall design impressions, how closely the plot of the movie relates to personal experiences, and other considerations. For this group, our attribute-based recommender algorithm likely will fail to provide reliable recommendations. In a practical context, this phenomenon implies that customers with unusual preferences cannot benefit from recommendations but instead suffer confusion or feel distracted by the inaccurate nature of recommendations. This effect can produce negative consequences for a retailer, such as decreased customer trust and loyalty, customer loss, or lowered revenues.

To counteract these negative effects, we suggest combining our attribute-based algorithm with an item-based CF algorithm that is known to perform reasonably well with respect to both prediction quality and the ability to provide explanations, with an influence explanation style, to users whose preferences are not based on product attributes (Claypool et al. 1999). It thus represents the next best alternative to a keyword explanation style that can be provided by our proposed CB method. In addition, some user characteristics might help identify whether a user is unusual, but they would need to be collected first, whether by observing user behavior, with surveys, or in exchange for prizes, for example. Furthermore, the influence explanation style should be informative for users with hard-to-operationalize preference structures, because it enables them to understand the commonalities among products that led to the recommendation and thereby infer latent factors that should be meaningful to them as they form preferences.

A classical item-based CF algorithm examines correlations (similarities) between the items a user liked most in the past (I_u^{best}) and items that have not been rated by the user (I'). In the next step (Eq. (9)), the algorithm predicts ratings for each item from I' as a sum of the user's ratings for the items from I_u^{best}, weighted by their respective similarities to the item of interest.

$$r_{u,i} = \frac{\sum_{i' \in \hat{I}} sim\left(i, i'\right) \cdot r_{u,i'}}{\sum_{i' \in \hat{I}} \left|sim\left(i, i'\right)\right|} \tag{9}$$

We rely on a switching hybridization design, which generates predictions of future ratings for a user, according to the method that achieves the lowest root mean square error (RMSE) with the same holdout data. To determine the method with better performance, we use t-tests for paired samples. If a method exhibits a significantly lower prediction error on the holdout set, we use it to generate the recommendations. If the difference between the errors is not significant, we use the predictions of the model in Eq. (3), even if it produces greater errors than the item-based CF approach for the holdout set. With this decision rule, we exchange formal accuracy for the prospect of a more effective explanation. Such hybridization also allows for the generation of the best possible recommendations, accompanied by actionable explanations for all users of a RS.

Empirical Study

To demonstrate the capability of our proposed method to estimate consumer preferences reliably, as well as its applicability in practical settings, we conducted an empirical study with

big data sets from two real-world RS. To assess the quality of the generated recommendations, we compared the accuracy of our method against that of several key recommendation techniques, including the most accurate technique published to date.[2]

Data

Two real-world data sets inform our empirical analysis. The first (Netflix) data set became publicly available in the context of the Netflix Prize[3] competition and has been employed extensively in recommender research. By using it, we ensure that our results are comparable with findings obtained through various other recommendation methods introduced in recently published investigations. The second data set is not publicly available; we obtained it from the movie recommendation website MoviePilot.[4] With this data set, we ensure that the comparison results are generalizable to recommendations other than those produced by Netflix and demonstrate the potential portability of our method to various product domains. Furthermore, we test the sensitivity of our method to the scale of the input data. Whereas in the Netflix data set, the ratings are represented on a 5-point scale (1 = "hated it" to 5 = "really liked it"), MoviePilot employs an 11-point scale that ranges from 0 ("hated this movie") to 10 ("my favorite movie"), in steps of .5, and it stores the ratings as tenfold values (i.e., a rating of 7.5 points is stored as 75).

Movies represent a typical focus of RS. For example, Netflix has reported that more than three-quarters of the movies that users have watched through its system reflect choices from among automated recommendations (Fiegerman 2013). By selecting this product class as our study focus, we can leverage the huge amount of existing research into relevant success factors (e.g., Carrillat, Legoux, and Hadida 2017; Clement, Wu, and Fischer 2014). In turn, we can better demonstrate the robustness of our proposed method in a domain marked by many attributes, often exceeding the number of data points (i.e., ratings per user) available for the estimation procedure. We derived the conceptual composition of preference-relevant movie attributes from extant research on success factors for motion pictures. To operationalize the attributes, we used data provided by IMDb.com and obtained the list of relevant movie stars, offscreen personnel, and firms with star power from InsideKino. Using these additional data enabled us to describe each movie by 374 attributes, as displayed in Table 2. After including two temporal ($p_{u,j}$, $\delta_{u,j}t$) and seven attribute-unrelated parameters (, b_u, $\alpha_u t$, b_i, $\beta_i t$, s_u, $\gamma_u t$) from Eq. (3), this number of movie attributes expands to $(374 \times 2 + 7) = 755$ model parameters to be estimated for each user.

To perform our tests, we reduced both data sets. We removed the six newest ratings from each user as a validation set for out-of-sample predictions and to compute the accuracy measures for different recommender algorithms. We drew another six ratings randomly from each user's rating profile to construct a holdout set. Next, we discarded consumers for whom insufficient data were available to generate holdout sets, as well as those for whom fewer than six ratings remained after isolating both holdout sets. The time range for the Netflix data set was 11 November, 1999 to 31 December, 2005; that for the MoviePilot data set was 19 August, 2006 to 4 April, 2008. We summarize their descriptive statistics in Table 3.

Measures and Benchmarks

Most studies of RS rely on two established accuracy measures: mean absolute error (MAE) and root mean squared error (RMSE). However, both these measures depend on the scale used to obtain ratings from consumers, and MoviePilot and Netflix rely on different rating scales. To ensure the comparability of predictions performed across the different data sets, we used normalized equivalents of these measures, which are insensitive to the rating scale (Goldberg et al. 2001; Herlocker et al. 2004).[5] The normalized mean absolute error (NMAE) and the normalized root mean squared error (NRMSE) are defined as follows:

$$NMAE = \frac{MAE}{r_{max} - r_{min}}, \text{ and} \tag{10}$$

$$NRMSE = \frac{RMSE}{r_{max} - r_{min}} \tag{11}$$

where r_{min} and r_{max} denote the minimum and the maximum ratings, respectively, of the rating scale of a particular recommendation system.

To provide an informative summary of the predictive accuracy of our proposed method, we examined the accuracy of pure user-based and item-based collaborative filters, each employing two variants of similarity measures, namely, Pearson's correlation coefficient and the cosine similarity metric. To assess these collaborative filters, we used a neighborhood of size $k = 50$, which provided the best accuracy of all examined data sets in our preliminary analyses. We also examine the database of product attributes for pairwise correlations. From each pair of attributes with high correlates, defined by $r_{ji} \geq .90$, we eliminated the attribute that exhibited lower variance in the data set.

[2] Although some proprietary algorithms may produce more accurate predictions, to the best of our knowledge, the Netflix Prize winner remains the most accurate published algorithm, to date.

[3] In 2009, Netflix awarded $1 million to the first research team to make successful, substantial improvements to the performance of its recommender algorithm, measured by RMSE (Koren et al., 2009).

[4] For this data set, we obtained full access to various information that could influence rating data, such as changes in the labels of a rating scale or updates to consumer interfaces, which was not available in the Netflix data (Koren 2009). Furthermore, Netflix has only released a subset of its rating data. In contrast, the MoviePilot data set is a complete listing of all ratings provided to the MoviePilot recommender system by its users.

[5] Other benchmark approaches such as popularity sorting (e.g., bestsellers) might be interesting but would need survey-based experiments to evaluate. In online retail stores, such approaches can easily be combined. For example, customers might select bestsellers as a basic filter and receive a list of the most popular products, with a personal prediction including explanations for each one.

Table 2
Preference-relevant movie attributes.

Attribute	Operationalization	Number of static parameters	Source
Star actors		133	InsideKino
Certification		29	IMDb
Country of origin		38	IMDb
Directors		106	InsideKino
Genre	Binary	26	IMDb
Language		22	IMDb
Producers		4	InsideKino
Production companies		6	InsideKino
Screenwriters		5	InsideKino
Admissions		1	IMDb
Box-office gross		1	IMDb
Budget	Metric	1	IMDb
Movie length		1	IMDb
Year of production		1	IMDb
Total:		374	

Table 3
Descriptive statistics for the data sets.

	Netflix data set			MoviePilot data set		
	Training set	Operation holdout	Validation set	Training set	Operation holdout	Validation set
General characteristics						
Number of ratings	93,170,314	2,570,310	2,570,310	1,140,577	47,610	47,610
Number of users	428,385	428,385	428,385	7,935	7,935	7,935
Number of movies	16,543	16,241	16,212	12,246	5,052	5,037
Ratings per user						
Min	8	6	6	6	6	6
Max	16,419	6	6	6,535	6	6
Mean	217	6	6	143	6	6
Median	101	6	6	59	6	6
SD	304.50	0	0	250.16	0	0
Ratings per movie						
Min	2	1	1	1	1	1
Max	213,367	15,816	12,354	4,543	802	677
Mean	5,623	158	158	93	9	9
Median	544	20	18	12	3	2
SD	16,305.89	624.28	603.76	262.90	36.23	35.7161
Ratings per day						
Min	5	1	1	1	1	1
Max	703,924	27,936	17,202	56,194	3,413	3,629
Mean	42,631	1,283	1,242	2,120	106	104
Median	15,167	38	61	1,293	50	52
SD	53,378.06	3,423.97	2,820.47	3,451.80	201.44	206.64

By applying these multicollinearity procedures, we removed 46 parameters, leaving 708 parameters to be estimated.

To assess the relative accuracy improvements achieved through different algorithms, we introduced two benchmarks. The first is a simple heuristic that predicts that the global average of a data set is the value of all future ratings for all users. The second benchmark is the Netflix Prize winner algorithm, which achieved an RMSE of .87120 and NRMSE of .21780 (Bell et al. 2009), an improvement of more than 10% relative to the RMSE of the original Netflix algorithm. Thus, it represents the most accurate recommendation algorithm currently known. In other words, no single or hybrid recommender, including switching hybrids, or any other methods such as hierarchical Bayes, demonstrates better rating predictions. This upper-level benchmark is informative for assessing the accuracy of our recommendation method.

Results

In this section, we address two questions: How well does our proposed method predict future user ratings, and what proportion of users receive recommendations accompanied by explanations in a keyword style?

Predictive Accuracies

We summarize the results of the prediction runs of different algorithms in Table 4. The results of both data sets are similar and consistent. We describe the accuracy of our proposed method in the bottom three rows of this table. Specifically, in the *Estimation step* we detail the predictions of Eq. (3), using parameter values obtained through the estimation portion of our algorithm. Then in the *Optimization step*, we specify the accuracy of the predictions for the same model, using optimized parameter values. Finally, the *Hybrid step* row indicates the accuracy of the predictions obtained by hybridizing our optimized solution with item-based collaborative filtering. Relative to the estimation step, the relative performance increases in the Netflix data set are .15% NMAE and .47% NRMSE for the optimization step, then an additional 9.29% NMAE and 9.41% NRMSE for the hybridization step. The values for the MoviePilot data set are similar, with .13% NMAE and .43% NRMSE for the optimization step and 10.86% NMAE and 14.53% for the hybridization step.

All these recommendation methods outperform the global average benchmark. Among collaborative methods, an item-based method using Pearson's similarity metric outperforms other methods with respect to NMAE but is only second-best with respect to NRMSE. Conversely, the user-based approach that adopts Pearson's similarity metric is the best collaborative approach with respect to NRMSE and the second-best approach with respect to NMAE. Despite some alternation in these two methods, the rank order of the different prediction methods with respect to accuracy remains generally consistent for both data sets. We interpret this finding as an indicator of the generalizability of our results.

Although the predictions of our attribute preference model in both the estimation and optimization steps exhibit significant accuracy improvements of more than fifteen percent compared with the global average estimator, this model is not the most accurate approach. The results from the optimization step also do not differ substantially from those achieved through the estimation step; we find only marginal improvements in both NMAE and NRMSE. However, the improvement in NRMSE is approximately four times greater than the improvement in NMAE. This result indicates that the optimized part-worth values reduce the magnitude of prediction errors in specific cases, instead of reducing the errors of all predictions.

Our proposed hybrid method outperforms all other methods and even the benchmark of the Netflix Prize winner algorithm, achieving a more than five percent improvement over the latter, though neither of the hybrid method's individual components reached this bottom-level error benchmark. This dramatic accuracy improvement merits closer examination. In Table 5, we summarize the distribution parameters of the absolute error after the optimization step.

For both data sets, our algorithm exhibits relatively high positive kurtosis values (greater than two) and a relatively low standard deviation (relative to the mean error). Therefore, most error values are concentrated around a particular point, instead of being spread across a wide interval. The analysis of the quantiles of the two distributions reveals that these error distributions are positively skewed. The peak and positive skew of the distributions also can be confirmed; the prediction error's standard deviation around the mean is lower than the RMSE values ($RMSE_{Netflix} = .90760$, $RMSE_{MoviePilot} = 24.17754$). Furthermore, the absolute prediction error is lower than the standard deviation, exceeding it only in approximately 30% of the examined cases. The error measures appear to be produced primarily by a few points with large deviations, rather than many points with nearly equal deviations. In combination, these findings offer evidence that our model generally predicts consumer ratings accurately but is inaccurate for approximately 30% of cases.

For users whose preferences are not predicted accurately by our proposed model, the magnitude of the prediction errors ranges from approximately 25% to 100% of the rating interval. To identify the source of these errors, we inspected the ratings for which our algorithm produced large errors and found they consistently linked to the same group of users. However, we could not find patterns that would permit an a priori identification of these users with high prediction errors. That is, the users do not exhibit any noticeable patterns with respect to the source data, such as fewer ratings or a tendency to rate specific movies. Such particular features would enable us to differentiate them from the users our algorithm assessed accurately. The only explanation we can devise for this phenomenon is that these "problematic" users form their movie preferences on the basis of information that is not captured by the preference function in Eq. (3). This observation aligns with our suggestion that the attribute-based preference model cannot capture movie preferences for some users; this inability to represent the preferences of certain users with the attribute-based model alone motivates the use of hybridization.

To further justify our hybridization method, we performed the Kolmogorov–Smirnoff test of the equality of distribution functions. The results revealed that the error distribution for item-based CF significantly differed from that of the attribute-based preference model ($< .01$ for both data sets). Both approaches produced unequal errors for most consumers on the single-consumer level ($p < .10$ in t-tests of the equality of means). These results confirm that the two approaches capture different types of variance in consumer ratings, and each approach is well suited for describing the preference formation of a different type of consumer. The hybridization scheme that leads to a better prediction, relative to the individual predictions offered by both approaches, is a sensible technique that generates substantial improvements in predictive accuracy.

Explanation Styles

Although our hybrid method cannot ensure that provided explanations appear in the most detailed style for all users, all users at least receive explanations in one of the two most effective styles. The attribute-based preference model of Eq. (3) cannot capture the preference structure of certain users, who form their preferences on the basis of factors other than movie attributes. Thus, recommendation explanations that cite movie attributes are not informative and do not increase the effective-

Table 4
Prediction accuracies of different algorithms, Netflix and MoviePilot data sets.

Category	Algorithm	Netflix data set							MoviePilot data set					
		NMAE	Rank (NMAE)	Improve-ment to global average (NMAE)	NRMSE	Rank (NRMSE)	Improve-ment to global average (NRMSE)	Improve-ment to Netflix prize winner (NRMSE)	NMAE	Rank (NMAE)	Improve-ment to global average (NMAE)	NRMSE	Rank (NRMSE)	Improve-ment to global average (NRMSE)
Benchmark methods	Global average	.22815	#8	.00%	.27725	#9	.00%	−27.30%	.21555	#8	.00%	.26345	#8	.00%
	Netflix winner	[a]	[a]	[a]	.21780	#2	21.4%	.00%	[a]	[a]	[a]	[a]	[a]	[a]
Collaborative filtering methods	User-based, Pearson	.16985	#4	25.55%	.21980	#4	20.72%	−.92%	.16921	#3	21.50%	.22158	#2	15.89%
	User-based, Cosine	.17387	#5	23.79%	.22376	#6	19.29%	−2.74%	.17373	#5	19.40%	.22551	#5	14.40%
	Item-based, Pearson	.16932	#2	25.79%	.21929	#3	20.91%	−.68%	.16807	#2	22.03%	.22174	#3	15.83%
	Item-based, Cosine	.16978	#3	25.58%	.21994	#5	20.67%	−.98%	.17214	#4	20.14%	.22521	#4	14.52%
Proposed method	Estimation step	.17680	#7	22.51%	.22797	#8	17.77%	−4.67%	.18187	#7	15.63%	.24283	#7	7.83%
	Optimization step	.17653	#6	22.63%	.22690	#7	18.16%	−4.18%	.18164	#6	15.73%	.24178	#6	8.23%
	Hybridization step	.16013	#1	29.81%	.20555	#1	25.86%	5.62%	.16192	#1	24.88%	.20665	#1	21.56%

Notes: In the "Rank" columns, lower rank numbers indicate greater accuracy.

[a] The Netflix Prize winning team (Bell et al. 2009) did not report the MAE or provide enough details to enable a replication with our MoviePilot data set; we derived the NRMSE from the reported RMSE.

Table 5
Distribution parameters for absolute prediction error of the optimization step.

Data set	Min	Max	Mean	SD	Mode	Kurtosis	SE of Kurtosis	25th percentile	50th percentile	75th percentile
Netflix	0	5	.706	.624	0	2.527	.018	.2315	.5368	1.44
MoviePilot	0	100	18.16	16.36	0	2.434	.022	6.03	13.60	25.48

ness of these users' choices. Because the item-based component of our hybrid approach substantially increases the predictive accuracy for these users, it appears to capture the "correct" aspects of their rating variance. An influence-based explanation style that emphasizes similarities among movies should be more informative and effective for them. For example, our approach might generate a keyword explanation, such as: "You will probably like the movie *Avengers: Endgame* because some of your favorite actors (*Scarlett Johansson* and *Chris Evans*) are in it, and because the movie belongs to your favorite genres (*Action*, *Adventure*, and *Sci-Fi*)." An influence explanation instead might read, "You will probably like the movie *Avengers: Endgame* because you liked *Avengers: Infinity War* and *Avengers: Age of Ultron*."

Table 6 summarizes the users in each data set who receive each particular explanation with the hybrid prediction method. These results are consistent for both data sets; we observe no substantial differences. Thus, our method scales well to big data. If the hybrid method were applied to these data sets, item-based CF and its associated influence explanation style would be employed for approximately 34% of the users. The majority of the users (66%) would receive recommendation justifications in the most detailed format, namely, the keyword explanation style.

Conclusion

Key Findings

We develop and test a novel, hybrid recommendation method that balances two concurrent aims and is inherently capable of providing both accurate product recommendations and explanations of the reasoning underlying these recommendations, in a way that is informative, understandable, and sensible, reflecting the terms the users themselves would use to describe their preferences. The accuracy of a non-hybrid RS is not better than benchmark approaches if the same method were applied to all users (as detailed in our Estimation and Optimization Step sections). This outcome changes with hybridization though. Our analyses confirm that consumers form preferences in different ways. Providing distinct recommendations for each group of consumers, using a method that captures their preferences appropriately, can substantially increase the predictive accuracy and credibility of a RS—and thus its value.

Furthermore, providing explanations of these recommendations that reflect the way each consumer thinks about the recommended products contributes to the quality and effectiveness of users' choices and increases their satisfaction. For the benefits of explanations to be realized, they must be understandable and actionable—traits best fulfilled by a keyword explanation style, which emphasizes the product attributes important to the user, or an influence explanation style, which cites the most influential products that produced the recommendation.

Implications

Managers should implement automated explanation tools in their RS, tightly coupled with the recommender algorithm, to increase consumer comprehension. In most cases, managers probably already have data about the particular product attributes in their online shop. If not, they can link them to their offers relatively easily, using insights from other databases (e.g., IMDb.com for movies, IGDb.com for video games). If recommendations are produced differently and appropriately, their explanations can reflect the underlying process of recommendation generation and emphasize the aspects that are most relevant to the consumer's own decision making. This approach also can increase a consumer's choice effectiveness and compensate for algorithmic prediction errors by allowing consumers to assess the quality and suitability of recommendations before they make choices. Explanations of recommendations provide additional decision-supporting information that enables consumers to address the decision context and evaluate other, fine-grained implications of their decisions.

In turn, RS providers should seek further understanding of the criteria that users employ to reach their decisions. Instead of one algorithm for all recommendations, optimal RS will handle users in individually tailored fashion. Specifically, hybrid systems should comprise different methods that reflect the decision-making processes of individual consumers. As our study shows, such RS can resolve recommendation issues adequately.

Further Research

Our proposed method can elicit the attribute-based preferences of individual consumers, even in settings in which the number of preference-relevant product attributes regularly exceeds the number of user ratings available for estimating a user's part-worths. This method employs auxiliary regressions that estimate one regression parameter at a time and relies on the properties of data sets to correct the estimates for omitted variable bias and multicollinearity, then optimize them for further reductions of prediction errors. Although it performs well in a RS setting, we encourage further research to expand applications to other problems, such as developing new products or estimating effects other than consumer preferences (e.g., other types of regressions). Moreover, we encourage explorations of multiplicative models that can account for non-linear attribute preference functions and temporal changes in user preferences.

Table 6
Explanation styles provided to users.

	Netflix data set		MoviePilot data set	
Explanation style	Number of users	Percentage of users	Number of users	Percentage of users
Keyword	290,146	67.73%	5,194	65.31%
Influence	138,239	32.27%	2,759	34.69%
Total	428,385	100%	7,953	100%

Regularization techniques, which also might be used to estimate user- and item-related effects, do not cut off based on a *p*-value as in our approach, but their procedure requires setting the value of the regularization parameter lambda. Moreover, some product attributes, such as certain actors and directors in our example, could be classified. Continued research could try to improve the performance of the algorithm with these alternative techniques. Another interesting question for further research pertains to options for combining our hybrid algorithm with other approaches. For example, customers of online retail stores might be interested in a current bestseller list of a selected category; perhaps this list could offer personalized recommendations, with explanations, for each entry. Measures of customers' trust in and perceptions of the usability and value of such combined approaches would provide helpful insights. We also suggest continued explorations of other uses for RS data and related approaches in other applications. Marketers might use preference data from RS to profile their customer bases, in support of market segmentation, product development, firm stock optimization, product bundling, marketing collateral, or direct mailing efforts. Recommender algorithms conceivably might analyze scanner or app activity data, which have great promise for optimizing loyalty programs.

Finally, we encourage researchers to devote greater attention to RS as instruments of the marketing mix, to define their roles, properties, potentials, capabilities, values, advantages, problems, and consequences for businesses in general and for marketing-related initiatives in particular.

Acknowledgements

The authors thank Thorsten Hennig-Thurau, Mark B. Houston, and Denis Rechkin for intellectual support, Tobias Bauckhage for providing the data for the tests, and the German Research Foundation (DFG) for its financial support of this project.

Appendix A. Details of Our Approach to Counteract the Omitted Variable Bias

To illustrate the rationale for this approach, consider an example: Suppose that the true regression model is expressed as Eq. (A1), but we omit the relevant variable X_2 and thus fit the model in Eq. (A2):

$$Y_n = \beta_0 + \beta_1 X_{1n} + \beta_2 X_{2n} + \varepsilon_n \tag{A1}$$

$$Y_n = \alpha_0 + \alpha_1 X_{1n} + \vartheta_n \tag{A2}$$

In the underspecified model in Eq. (A2), the expected value $E(\hat{\alpha}_1)$ of the slope of X_1 equals the sum of two quantities: the true value of the slope β_1 that would be obtained from the regression of the true model in Eq. (A1) and the product of the true value of the slope β_2 of the omitted variable X_2 and the slope b_{21} of the auxiliary regression of X_2 on X_1, or formally:

$$E(\hat{\alpha}_1) = \beta_1 + \beta_2 b_{21} \tag{A3}$$

where b_{21} is the slope in the auxiliary regression of X_2 on X_1. Here, $\hat{\alpha}_1$ is biased unless β_2 and/or b_{21} equals 0, which would imply that X_2 has no effect on Y or that X_1 and X_2 are uncorrelated. Thus, the first step for determining the bias of an estimate is examining the correlations between the variables. If we find no correlations, the estimate of the corresponding parameter and its variance is unbiased; if X_1 and X_2 are correlated, the estimate of $\hat{\alpha}_1$ is biased. In this case, the bias can be corrected by fitting two underspecified models from Eq. (A2) and two auxiliary regressions from Eq. (A4) of X_1 on X_2, and vice versa:

$$X_1 = b_{10} + b_{12} X_2 \,;\; X_2 = b_{20} + b_{21} X_1 \tag{A4}$$

The slopes b_{21} and b_{12} from these auxiliary regressions then can be substituted into Eq. (A3), which produces a system of two equations with two unknowns:

$$\hat{\alpha}_1 = \beta_1 + \beta_2 b_{21} \,;\; \hat{\alpha}_2 = \beta_2 + \beta_1 b_{12} \tag{A5}$$

This system can be algebraically solved for β_1 and β_2, producing the bias-corrected estimates of the effects of interest:

$$\beta_1 = \frac{\hat{\alpha}_1 - \hat{\alpha}_2 b_{21}}{1 - b_{12} b_{21}} \,;\; \beta_2 = \hat{\alpha}_2 - \beta_1 b_{12} \tag{A6}$$

The next step is to correct the variance estimates for β_1 and β_2, which is required because the variance estimates are involved in the calculation of the *t*-value of Student's t-test. That is, biased variance estimates lead to misleading conclusions about an effect's significance and confidence limits. To counteract this issue, we can recalculate the variance:

$$var(\beta_i) = \frac{\sigma^2}{\sum x_{in}^2} VIF = \frac{\sum \vartheta_n^2 / df}{\sum x_{in}^2 \left(1 - z_{ij}^2\right)} \tag{A7}$$

where $VIF = 1/\left(1 - z_{ij}^2\right)$ is the variance inflation factor, and z_{ij}^2 is the maximum of the multiple coefficients of determination of the regression of X_i on the other covariates. However, prior to completing this recalculation, it is necessary to obtain the value of the residual sum of squares, $\sum \vartheta_n^2 = \sum (Y_n - \hat{Y}_n)^2$, of the "true" OLS model in Eq. (A1). With the bias-corrected values

of β_1 and β_2 and the definition of the constant term from Eq. (A8), we can calculate $\hat{Y}_n$ and thereby determine the value of $\sum \vartheta_n^2$:

$$\hat{\beta}_0 = \bar{Y} - \hat{\beta}_1 \bar{X}_1 - \hat{\beta}_2 \bar{X}_2 \tag{A8}$$

The number of degrees of freedom for OLS is the number of data points minus the number of regressors minus one. It is set to one in cases in which there are more predictors than data points, to avoid an infinite value for the variance. We thus have all the information needed to recalculate $var\left(\hat{\beta}_i\right)$ from Eq. (A7). In the next step, the bias-corrected t -value and confidence limits can be obtained from their definitions, using bias-corrected variance estimates. Finally, we test for significance using the corrected t -values. This procedure requires at least three ratings per user to estimate auxiliary regressions with valid standard errors. For further details, see Web Appendix D in Supplementary material.

Appendix B. Supplementary data

Supplementary material related to this article can be found, in the online version, at doi:https://doi.org/10.1016/j.jretai.2020.01.001.

References

Adomavicius, Gediminas and Alexander Tuzhilin (2005), "Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions," *IEEE Transactions on Knowledge and Data Engineering,* 17 (6), 734–49.

Aksoy, Lerzan, Paul N. Bloom, Nicholas H. Lurie and Bruce Cooil (2006), "Should Recommendation Agents Think Like People?," *Journal of Service Research,* 8 (4), 297–315.

André, Quentin, Ziv Carmon, Klaus Wertenbroch, Alia Crum, Douglas Frank, William Goldstein, Joel Huber, Leafvan Boven, Bernd Weber and Haiyang Yang (2018), "Consumer Choice and Autonomy in the Age of Artificial Intelligence and Big Data," *Customer Needs and Solutions,* 5 (1–2), 28–37.

Avery, Christopher and Richard Zeckhauser (1997), "Recommender Systems for Evaluating Computer Messages," *Communications of the ACM,* 40 (3), 88–9.

Bell, Robert, Jim Bennett, Yehuda Koren and Chris Volinsky (2009), "The Million Dollar Programming Prize," *IEEE Spectrum,* 46 (5), 28–33.

Bennet, James and Stan Lanning (2007), "The Netflix Prize," *Proceedings of KDD Cup and Workshop,*,. August 12, 2007

Bilgic, Mustafa and Raymond J. Mooney (2005), "Explaining Recommendations: Satisfaction vs. Promotion," *IUI'05 Beyond Personalization Workshop,*, 13–8.

Billsus, Daniel and Michael J. Pazzani (2000), "User Modeling for Adaptive News Access," *User-Modeling User-Adapted Interaction,* 10 (2–3), 147–80.

Bleier, Alexander and Maik Eisenbeiss (2015), "The Importance of Trust for Personalized Online Advertising," *Journal of Retailing,* 91 (3), 390–409.

Bodapati, Anand V. (2008), "Recommendation Systems With Purchase Data," *Journal of Marketing Research,* 45 (1), 77–93.

Burke, Robin (2002), "Hybrid Recommender Systems: Survey and Experiments," *User Modeling and User-Adapted Interaction,* 12 (4), 331–70.

Carrillat, François A., Renaud Legoux and Allègre L. Hadida (2017), "Debates and Assumptions About Motion Picture Performance: A Meta-Analysis," *Journal of the Academy of Marketing Science,*, 1–27.

Chen, Li (2009), "Adaptive Tradeoff Explanations in Conversational Recommenders," *Proceedings of the third ACM Conference of Recommender Systems – RecSys'09,*, 225–8.

Chung, Tuck Siong, Roland T. Rust and Michel Wedel (2009), "My Mobile Music: An Adaptive Personalization System for Digital Audio Players," *Marketing Science,* 28 (1), 52–68.

Claypool, Mark, Anuja Gokhale, Tim Miranda, Paul Murnikov, Dmitry Netes and Matthew Sartin (1999), "Combining Content-Based and Collaborative Filters in an Online Newspaper," *Proceedings of ACM SIGIR Workshop on Recommender Systems'99,*, 1–8.

Clement, Michel, Steven Wu and Marc Fischer (2014), "Empirical Generalizations of Demand and Supply Dynamics for Movies," *International Journal of Research in Marketing,* 31 (2), 207–23.

Cramer, Henriette, Vanessa Evers, Satyan Ramlal, Maartenvan Someren, Lloyd Rutledge, Natalia Stash, Lora Aroyo and Bob Wielinga (2008), "The Effects of Transparency on Trust in and Acceptance of a Content-Based Art Recommender," *User Modeling and User-Adapted Interaction,* 18 (5), 455–96.

Fiegerman, Seth (2013), *Netflix Knows You Better Than You Know Yourself,* Mashable. http://mashable.com/2013/12/11/netflix-data/#WJk0_7cPiEql

Fitzsimons, Gavan J. and Donald R. Lehmann (2004), "Reactance to Recommendations: When Unsolicited Advice Yields Contrary Responses," *Marketing Science,* 23 (1), 82–94.

Fleder, Daniel and Kartik Hosanagar (2009), "Blockbuster Culture's Next Rise or Fall: The Impact of Recommender Systems on Sales Diversity," *Management Science,* 55 (5), 697–712.

Gedikli, Fatih, Dietmar Jannach and Mouzhi Ge (2014), "How Should I Explain? A Comparison of Different Explanation Types For Recommender Systems," *International Journal of Human-Computer Studies,* 72 (3), 367–82.

Gershoff, Andrew D., Ashesh Mukherjee and Anirban Mukhopadhyay (2003), "Consumer Acceptance of Online Agent Advice: Extremity and Positivity Effects," *Journal of Consumer Psychology,* 13 (1), 161–70.

Goldberg, Ken, Theresa Roeder, Dhruv Gupta and Chris Perkins (2001), "Eigentaste: A Constant Time Collaborative Filtering Algorithm," *Information Retrieval,* 4 (2), 133–51.

Gujarati, Damodar N. (2009), *Basic Econometrics,* 5th ed. Boston: McGraw Hill Education.

Hennig-Thurau, Thorsten, André Marchand and Paul Marx (2012), "Can Automated Group Recommender Systems Help Consumers Make Better Choices?," *Journal of Marketing,* 76 (5), 89–109.

Herlocker, Jonathan L., Joseph A. Konstan and John Riedl (2000), "Explaining Collaborative Filtering Recommendations," *Proceedings of the 2000 ACM Conference on Computer Supported Cooperative Work – CSCW'00,*, 241–50.

Herlocker, Jonathan L., Joseph A. Konstan, Loren G. Terveen and John T. Riedl (2004), "Evaluating Collaborative Filtering Recommender Systems," *ACM Transactions of Information System (TOIS),* 22 (1), 5–53.

Jannach, Detmar, Markus Zanker, Alexander Felfernig and Gerhard Friedich (2011), *Recommender Systems: An Introduction,* New York: Cambridge University Press.

Jolliffe, Ian T. (2014), *Principal Component Analysis,* Wiley. Statistics Reference Online.

Konstan, Joseph A., Bradley N. Miller, David Maltz, et al. (1997), "GroupLens: Applying Collaborative Filtering to Usenet News," *Communications of the ACM,* 40 (3), 77–87.

Koren, Yehuda (2009), "Collaborative Filtering with Temporal Dynamics," *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining – KDD'09,*, 447–56.

Koren, Yehuda, Robert Bell and Chris Volinsky (2009), "Matrix Factorization Techniques for Recommender Systems," *Computer,* 42 (6), 30–7.

Lee, Dokyun and Kartik Hosanagar (2019), "How do Recommender Systems Affect Sales Diversity? A Cross-Category Investigation via Randomized Field Experiment," *Information Systems Research,* 30 (1), 239–59.

Linden, Greg, Brent Smith and Jeremy York (2003), "Amazon.com Recommendations: Item-to-Item Collaborative Filtering," *IEEE Internet Computing,* 7 (1), 76–80.

Pazzani, Michael J. and Daniel Billsus (2007), "Content-Based Recommendation Systems," *Lecture Notes in Computer Science,* 4321, 325–41.

Press, William H., Saul A. Teukolsky, William T. Vetterling and Brian P. Flannery (2007), *Numerical Recipes: The Art of Scientific Computing,* New York: Cambridge University Press.

Ricci, Frabcesco, Lior Rokach, Bracha Shapira and Paul B. Kantor (2015), *Recommender Systems Handbook,* 2nd ed. New York: Springer.

Sarwar, Badrul, George Karypis, Joseph Konstan and John Riedl (2001), "Item-Based Collaborative Filtering Recommendation Algorithms," in *Proceedings of the 10th International Conference on World Wide Web, ACM*, 285–95.

Syam, Niladri B. and Nanda Kumar (2006), "On Customized Goods, Standard Goods, and Competition," *Marketing Science,* 25 (5), 525–37.

Symeonidis, Panagiotis, Alexandros Nanopoulos and Yannis Manolopoulos (2008), "Providing Justifications in Recommender Systems," *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans,* 38 (6), 1262–72.

Tintarev, Nava and Judith Masthoff (2007), "Effective Explanations of Recommendations," *Proceedings of the 2007 ACM Conference on Recommender Systems – RecSys'07,*, 153–6.

——— and ——— (2012), "Evaluating the Effectiveness of Explanations For Recommender Systems," *User Modeling and User-Adapted Interaction,* 2012, 399–439.

Vincent, James (2019), *AI Systems Should be Accountable, Explainable, and Unbiased, Says EU,* The Verge. https://www.theverge.com/2019/4/8/18300149/eu-artificial-intelligence-ai-ethical-guidelines-recommendations

Zhang, Shuai, Lina Yao, Shuai Zhang and Lina Yao (2019), "Deep Learning Based Recommender System: A Survey And New Perspectives," *ACM Computing Surveys,* 52 (1), 5 1–5:38.