Evaluating Recommender Systems with User Experiments

Bart P. Knijnenburg, Martijn C. Willemsen

Abstract Proper evaluation of the user experience of recommender systems requires conducting user experiments. This chapter is a guideline for students and researchers aspiring to conduct user experiments with their recommender systems. It first covers the theory of user-centric evaluation of recommender systems, and gives an overview of recommender system aspects to evaluate. It then provides a detailed practical description of how to conduct user experiments, covering the following topics: formulating hypotheses, sampling participants, creating experimental manipulations, measuring subjective constructs with questionnaires, and statistically evaluating the results.

1 Introduction

Traditionally, the field of recommender systems has evaluated the fruits of its labor using metrics of algorithmic accuracy and precision (see Chapter **??** for an overview of recommender systems evaluation practices). Netflix organized a million-dollar contest for just this goal of improving the accuracy of its movie recommendation algorithm [7]. In recent years, however, researchers have come to realize that the goal of a recommender system extends well beyond accurate predictions; its primary real-world purpose is to provide personalized help in discovering relevant content or items [72].

This has caused two important changes in the field. The first change was incited by McNee et al. [83] who argued that "being accurate is not enough" and that one should instead "study recommenders from a user-centric perspective to make them not only accurate and helpful, but also a pleasure to use" (p. 1101). McNee et al.

Bart P. Knijnenburg
UC Irvine, CA, USA, e-mail: bart.k@uci.edu

Martijn C. Willemsen
Eindhoven University of Technology, Eindhoven, The Netherlands, e-mail: m.c.willemsen@tue.nl

suggest broadening the scope of research regarding the *outcomes* of the evaluation beyond accuracy measures. This suggestion has spawned a research area that evaluates recommender systems in online user experiments with user-centric evaluation metrics that span behaviors (e.g. user retention and consumption) as well as attitudes (e.g. usability, choice satisfaction, and perceived usefulness; cf. [67, 95]).

The second change is a broadening of the scope of research regarding the *system aspects* to investigate beyond just the algorithm of the recommender. In essence, recommender systems apply algorithms on user input with the goal of providing some kind of personalized output. This means that aside from the algorithm, there are two important interactive components to any recommender: the mechanism through which users provide their input, and the means by which they receive the system's output. Realizing the importance of these interactive components, McNee et al. [84] suggested that researchers should put more focus on the "Human-Recommender Interaction" and investigate these interactive components. Moreover, in his RecSys 2009 keynote Martin emphasized the importance of this endeavor: he argued that the interactive components of a recommender account for about 50% of its commercial success, while he provocatively estimated that the algorithm accounts for only 5% [81]. Indeed, research has shown that the preference elicitation mechanism and the presentation of recommendations have a substantial impact on users' acceptance and evaluation of recommender systems as well as their usage behavior (cf. [19, 67, 96]).

These two changes have gradually evolved the field to take broader perspective on the user experience of recommender systems [72]. However, the majority of current research on recommender systems is still primarily focused on creating better algorithms, and conducts offline machine learning evaluations instead of "live" user experiments. The contribution of that research is thus limited to claims about algorithmic accuracy and precision; without performing any user-centric evaluations it is difficult to extend these claims to the more user-centric objective of recommender systems: giving users a pleasant and useful personalized experience.

Proper evaluation of the user experience of a recommender system requires conducting a *user experiment*,[1] either in the form of a lab experiment or a randomized field trial (which includes—but also extends beyond—conventional A/B tests). This chapter of the Recommender System Handbook is meant as a guideline for students and researchers aspiring to conduct user experiments with their recommender systems, as well as for editors and reviewers of conferences and journals to evaluate manuscripts. To this end, this chapter will provide both theoretical and practical guidelines. The theoretical part starts with the description of the Knijnenburg et al. [67] User-Centric Evaluation Framework for Recommender Systems. We subsequently use this framework to highlight aspects of recommenders and their users that could be the object of study. We outline what has already been tested, and where gaps in the literature exist. In the practical part, we provide guidelines regarding all

[1] We use the term "user experiment" to denote the use of experimental conditions and formal measurement as a means of testing theories about users interacting with recommender systems. This as opposed to "user studies", which are typically smaller observational studies used to iteratively improve the usability of a recommender system.

the steps involved in setting up, conducting and analyzing user experiments. The framework will be used there to motivate and illustrate our practical guidelines.

This chapter is meant as a practical primer; a succinct yet comprehensive introduction to user experiments, motivated by numerous examples of published recommender systems studies. The reader who is serious about conducting user experiments is encouraged to continue their learning process beyond this chapter. To this effect we have listed a number of excellent textbooks in the conclusion of this chapter.

2 Theoretical Foundation and Existing Work

An essential part of conducting a good experiment is to have a good *research model* (or descriptive theory, cf. [53]) of how the aspects under evaluation interact (see Sect. 3.1). Such models are usually based on a synthesis of formal theories and existing research, identifying the unknown parameters, and formulating testable hypotheses regarding these parameters. To add some structure to the process of theory development, it is helpful to conceptualize the interaction between users and recommenders within a theoretical framework. Several of such frameworks exist (cf. [84, 95]), but we choose to structure this chapter around the Knijnenburg et al. [67] User-Centric Evaluation Framework for Recommender Systems.

2.1 Theoretical Foundation: The Knijnenburg et al. Evaluation Framework

The Knijnenburg et al. [67] framework consists of two levels (see Fig. 1). The top level is a middle range "EP type" theory[2] of how users experience an interactive information system. A middle range theory is a theory about human behavior that is applicable in a specific but reasonably generic situation (in this case: in using an interactive information system). An "EP type" theory is a theory that can be used to explain (E) the described behavior and to predict (P) how users would behave under specific circumstances. The theory that comprises the top level of the Knijnenburg et al. framework combines [3] existing theories of attitudes and behaviors [2, 3, 4, 37], technology acceptance [26, 116], and user experience [46, 47]. Specifically, it describes how users' subjective interpretation (Subjective System Aspects, or SSA) of a system's critical features (Objective System Aspects) influences their experi-

[2] See [45] for a taxonomy of different types of theory.

[3] Like Hassenzahl [46, 47], our framework describes the formation of experiences during technology use rather than the longer-term phenomenon of technology acceptance, but it extends this model to behavioral consequences using attitude-behavior theories [2, 3, 4, 37] (a theoretical structure that is prominent in technology acceptance models [26, 116]).

ence of (EXP) and interaction with (INT) a system. Note that the top level of the framework can potentially be applied beyond the field of recommender systems.

The lower level of the Knijnenburg et al. framework is a classification of recommender system related constructs under these higher level concepts (inspired by related analysis-type frameworks of recommender system aspects [84, 95, 122]). These constructs can be used to turn the top-level theory into models for specific recommender system evaluation studies. The combination of a top level theory and a lower level taxonomy makes our framework more actionable than [84] (because the EP type theory provides concrete suggestions for specific research hypotheses) and more generic than [95] (because the EP type theory is generative, which makes our framework more easily adaptable to new areas of recommender system research). The Knijnenburg et al. framework has been put to practice in several published and unpublished studies, so we will be able to illustrate many of our practical guidelines with examples from existing applications of this framework.

An updated version[4] of the Knijnenburg et al. [67] evaluation framework is displayed in Fig. 1. It represents the user-centric evaluation of recommender systems as six interrelated conceptual components:

Objective System Aspects (OSAs) As recommender systems are typically multi-faceted systems, their evaluation should be simplified by considering only a subset of all system aspects in each experiment. The Objective System Aspects

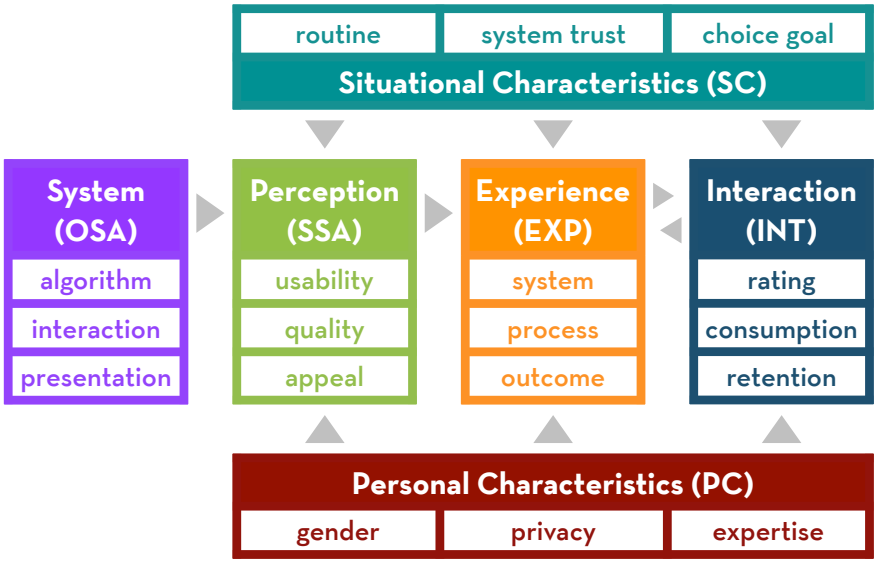

Fig. 1 An updated version of the User-Centric Evaluation Framework [67].

[4] The paths from Personal and Situation Characteristics to Subjective System Aspects were added to the original framework (as presented in [67]) based on insights from various experiments with the framework.

(OSAs) are the aspects of the system that are currently being evaluated. The algorithm can be considered as an OSA, but also the input (interaction) mechanisms (e.g. the rating scale used to provide feedback on recommendations) or output (presentation) mechanisms (e.g. the number of presented recommendations, or their layout).

Subjective System Aspects (SSAs) Although we are ultimately interested in the effects of OSAs on User Experience (EXP) and Interaction (INT), we need to consider Subjective System Aspects (SSAs) as mediating variables of these effects. SSAs are users' perceptions of the OSAs. SSAs are measured with questionnaires that participants are asked to complete after (or sometimes during) their interaction with the system (see Sect. 3.4). The measurement of SSAs is necessary because incremental advances in recommender system aspects (e.g. algorithms) are often small, and may go unnoticed. SSAs help establish whether users *perceive* a certain system aspect, independently of their *evaluation* of the aspect. For example, if an improved system does not lead to the expected increase in user satisfaction, the SSA "perceived recommendation quality" can be used to find out if users simply did not notice the improvement, or if they noticed it but did not really like it. SSAs mediate the effects of OSAs on EXP, thereby explaining how and why OSAs influence EXP, as well as increasing the robustness of this causal link.

User Experience (EXP) The User Experience factors (EXPs) are users' self-relevant evaluations of the qualities of the recommender system. User experience is also measured with questionnaires. Note that experience can relate to different aspects of system usage, namely the evaluation of the recommender system itself (e.g. perceived system effectiveness; system-EXP), the evaluation of the process of using the system (e.g. expressing preferences, and browsing or choosing recommended items; process-EXP), or the evaluation of the chosen items (e.g. choice satisfaction; outcome-EXP). It is important to make these distinctions, because different OSAs may influence different aspects of the experience.

Interaction (INT) The "final step" in the evaluation of a recommender system is the users' interaction with the system (INT). The interaction can be measured objectively by logging the users' clicks. Examples are: the number of recommendations inspected by the user, their rating feedback, and the time they spent using the recommender. Behavior grounds the subjective part of the evaluation in observable behavior. At the same time, the subjective components provide explanations for the (sometimes counterintuitive) observed behaviors.

Personal and Situational Characteristics (PCs and SCs) Although the main objective of most user experiments is to test the effects of OSAs on SSAs, EXPs and INTs, these outcomes can also be influenced by Personal Characteristics (e.g. domain knowledge; PCs) and Situational Characteristics (e.g. choice goals; SCs). PCs and SCs are typically measured by questionnaires[5], and since they are be-

[5] In some cases PCs and SCs can be inferred from user behavior, e.g. observing the click-stream can tell us the market segment a user belongs to [44]. SCs can also be manipulated, e.g. by priming users to approach the recommender with either a concrete or abstract mindset [71, 120]

yond the influence of the system they can be measured before users interact with the system.

The evaluation framework can be used as a conceptual guideline for developing hypotheses. It can answer questions like:

Which EXP aspects is this OSA likely to influence? For example, an improved algorithm may influence users' evaluation of the recommendations (outcome-EXP), while a new preference elicitation method is likely to influence the perceived effectiveness of the recommendation process (process-EXP). Both may impact users' satisfaction with the system itself (system-EXP).

Which SSAs can be used to explain these effects? For example, certain algorithms may produce more accurate recommendations, while other algorithms may increase the diversity of the recommendations. Both may increase user satisfaction, but for different reasons.

Which PCs and SCs may moderate these effects? For example, users' liking of accurate or diverse recommendations may depend on their choice goals (SC). The most suitable preference elicitation method may depend on users' domain knowledge (PC).

Like most theories [2, 3, 4, 26, 37, 116], the theoretical top level of the Knijnenburg et al. [67] evaluation framework is *generative*: experimenters should see the relationships between OSA, SSA, EXP, and INT as a blueprint for their own descriptive models, but define their own set of measurable constructs and manipulations that are tailored to their experiment. This way, the framework can help answer questions that are specifically relevant to the system under evaluation.

2.2 Overview of Existing User-Centric Work and Promising Directions

The main contribution of any recommender system user experiment is an empirical evaluation of how selected OSAs influence the user experience, possibly moderated by PCs and SCs. To aid the selection of interesting research topics, we provide a brief overview of OSAs that have been studied in the past, and some promising directions for future work. When writing a related works section for their own papers, researchers are advised to also consult other existing overviews of user-centric research in recommender systems, such as the following:

- Xiao and Benbasat [122] provide a thorough overview and synthesis of 47 empirical user-centric studies on what they call "recommendation agents". Their synthesis consists of a conceptual model that served as inspiration for the Knijnenburg et al. [67] framework. The authors recently updated their overview [123].
- Pu et al. [96] provide an overview of the state-of-the-art of user-centric recommender systems studies. Their synthesis consists of a number of practical design guidelines for recommender systems developers (see also Chapter **??**).

- Konstan and Riedl [72] put the rise of user-centric evaluation of recommender systems into a historical context. They focus on user-centric implications of technical aspects of recommender systems.

Here we discuss the most commonly researched OSAs of recommender systems. Envisioning a recommender system as a generic system that processes inputs to produce outputs, the main OSA categories are the input (preference elicitation), processing (algorithm) and output (recommendations and the presentation thereof). Our overview is meant for researchers who wish to evaluate the user experience of recommender systems. Researchers who wish to use recommender systems as a vehicle for researching aspects of human decision making are referred to Chapter **??** for a comprehensive overview.

2.2.1 Preference elicitation methods

The four most common methods recommender systems use to elicit preferences from users are rating scales, attribute weights, critiques, and implicit behavior. Rating scales are the most commonly employed method. They vary in granularity from binary (thumbs up/down), via the most common star ratings (5 stars or 10 half stars), to sliders (any number of steps). Research has shown that users behave differently depending on the used rating scale [42]. Users seem to prefer the 5-star and 10-half-star scales [15, 23, 28, 42, 106]. The more granular rating methods are more effortful, but also provide more information [60]. Regardless of the rating scale, user-ratings are often inaccurate [5, 100], and helping users with the rating task can increase their accuracy [87].

Preference elicitation via attribute weights originates from the field of decision analysis, where multi-attribute utility theory is used as a standard for rational decision-making [9]. Early work in this area shows that attribute-based recommenders result in better decisions and less effort compared to static browsing tools [48]. This benefit is moderated by domain knowledge: only experts are more satisfied with attribute-based recommenders and their outcomes; for novices, expressing preferences in terms of needs or examples tends to work better [65, 66, 98].

Another method to elicit preferences is example critiquing. In this method, users iteratively provide detailed feedback on example recommendations. Substantial user-centric work in this area (as summarized in [19]) shows that example critiquing systems save cognitive effort and increase decision accuracy. Moreover, aiding users by suggesting critiques seems to improve users' decision confidence [16]. On the other hand, Lee and Benbasat [77] show that a preference elicitation method that highlights trade-offs may increase users' trade-off difficulty.

A recommender system needs a certain number of ratings before it can produce accurate recommendations, but not all users may have rated that many items yet; this is the so-called "cold start problem". Implicit behavioral feedback such as browsing or purchase/consumption actions can be used to compute recommendations in such cases. In [67] we compared the use of explicit and implicit feedback to calculate recommendations. The results of this study showed that an implicit feed-

back recommender can provide higher-quality recommendations that result in a higher perceived system effectiveness and higher choice satisfaction. The results also showed that users perceived the explicit feedback-based recommendations to be more diverse, though, and diversity is another good quality of recommendation lists (cf. [120, 121, 126], see also Chapter **??**) The best solution is thus to create a hybrid system that uses both explicit and implicit feedback. Koren et al. [73] show that such hybrid recommenders are usually more accurate than their implicit and explicit counterparts (see also Chapter **??**). In [65] we show that hybrid recommenders are especially satisfying and effective for experts; for novices they seem to be too complex.

Another way to overcome the cold start problem is to encourage users to rate more items. Work on this topic shows that the best way to get users to rate more items is to show them the benefit of rating by presenting good recommendations early on in the interaction [33, 39, 68].

Future work could conduct a more comprehensive evaluation across the listed preference elicitation paradigms, or explore how the most suitable preference elicitation method depends not just on users' personal characteristics [65], but also on situational characteristics such as users' current mindset or choice goal.

2.2.2 Algorithms

As mentioned in the introduction, algorithms are often evaluated in an offline setting. More accurate algorithms are often assumed to result in higher quality recommendations and more effective systems, but this is not necessarily always the case. For example, McNee et al. [82] found that users rated their most accurate algorithm as least helpful, and Torres et al. [112] found that users were most satisfied with their least accurate algorithm. Despite the prevalent opinion that recommender systems research should move beyond offline evaluations to user-centric studies [72], surprisingly few research papers about new algorithmic solutions test the effect of the proposed algorithm on users' satisfaction (some exceptions are [25, 29, 99, 31]). Given the results of McNee et al. [82] and Torres et al. [112], we strongly suggest that algorithm developers test whether the accuracy improvements of their algorithms translate to a higher user satisfaction.

2.2.3 Recommendations and Their Presentation

The composition and presentation of the list of recommendations has a strong effect on the user experience. Choosing among top recommendations is a difficult task, and may lead to a phenomenon called "choice overload" [12]. Overcoming choice overload is one of the main challenges of research on the presentation of recommendation. Longer lists of recommendations may attract more attention [109], but are generally harder to choose from [6, 12]. Diversifying recommendations seems to be a good antidote against choice overload, because diversified lists are

attractive even when short [120, 121, 126]. In fact, non-personalized diversified lists can be as attractive as personalized recommendations [67]. A steady stream of research has considered algorithmic solutions to diversifying recommendations [1, 76, 115, 124, 125]. More research needs to be done on whether these algorithmic solutions indeed result in *perceptibly* more diverse recommendations, and on whether these recommendations reduce choice overload and increase user satisfaction.

The *layout* of the recommendations on the screen determines the amount of attention users pay to each recommendation. In a vertical list, users pay more attention to the first few items than to items lower down the list [12], but this decay is much less when using a grid layout [18]. In a grid layout, items in the top-left of the grid are taken to be the most relevant [57]. Chen and Tsoi [20] show that if recommendations are divided over two pages, the items on the second page get very few clicks. Comparing a list, grid and pie (circular) layout for recommendations, they find a slight user preference for the pie layout. This layout does however take up much more space on the screen.

In many commercial recommender systems the recommendations are organized into distinct *categories*. Chen and Pu [17] have developed a "Preference-Based Organization Interface" that uses categories as a basis for critiquing. In their system, the primary category has the user's top recommendations, and each other category explores a trade-off. Hu and Pu [52] show that this kind of categorization increases the perceived diversity of the recommendations. Beyond this, the categorization of recommendations has not received much attention in academic research but consumer research literature [85, 103] suggests that categorization structures the user's choice task, and helps to overcome choice overload.

Another challenge for recommender systems is to *explain* their recommendations (see [40, 41, 110] for an overview). Explanations can be based on the preferences of similar users (e.g. "this item was rated highly by users similar to you"), similar items (e.g. "this is similar to other items you liked"), or attributes/keywords of interest (e.g. "this has attributes you prefer"). Explanations can be presented textually (e.g. as a number, keyword, text or tag cloud) or visually (e.g. as a histogram or pie chart). Research has found that users like explanations [50], and that they increase users' understanding of the recommendation process [41, 117], their trust in the quality of the recommendations, and the competence and benevolence of the system [24, 36, 119] (more on credibility and trust can be found in Chapter **??**). This in turn increases their purchase intentions [118] and their intention to return to the system [94].

Which type of explanation works best? Research comparing different types of explanation strategies has found that explanations based on the preferences of similar users are persuasive: users tend to overestimate the quality of recommendations explained this way [10, 41, 50]. Item- and keyword-based explanations produce more accurate expectations [10, 41] and ultimately lead to more satisfaction [41, 108]. Finally, Pu and Chen demonstrate that carefully organizing the list of recommendations may also be perceived as an implicit explanation [94]. This type of explanation produces little perceived cognitive overhead.

Tintarev and Masthoff [111] explore the idea of personalizing explanations to the user. They show that users tend to like such personalized explanations, but that these may actually be less effective than generic explanations. Social recommenders that use a user's friends instead of anonymous nearest neighbors for recommendation purposes have an additional opportunity for explanation, as they can show how recommendations are linked to the preferences of the user's friends. In [62] we demonstrate that displaying such a "recommendation graph" increases the inspectability of the recommendations, and ultimately users' satisfaction with the system.

There is no doubt that explaining recommendations is beneficial for the user experience, because they help users to increase their understanding of the recommendation process. However, users can also use explanations to justify their choice among the presented recommendations, which could arguably reduce choice overload and increase their decision confidence (see Chapter **??**). We reiterate the conclusion by [72, 111] that future work should explore how explanations can help to reduce choice overload and otherwise improve users' decision-making.

Work on the presentation of recommendations generally considers variants of the conventional "Top-N" list of recommendations. Alternative uses of recommendations are becoming more prevalent in practice, though. Examples are "co-recommendations" ("Users who bought this also bought…" [89, 90]) and "smart defaults" (recommendations as default settings for yes/no or multiple-option decisions [61, 105]). The presentation of these types of recommendations has to date not been investigated in much detail.

3 Practical Guidelines

We now turn to the practical part of this chapter, where we provide guidelines regarding the different steps involved in recommender system user experiments. Sect. 3.1 (Research Model) deals with developing a research model and hypotheses for the experiment. Sect. 3.2 (Participants) discusses the recruitment of test users. Sect. 3.3 (Manipulations) covers the operationalization of hypotheses into different versions of the system and the process of randomly assigning participants to these versions. Sect. 3.4 (Measurement) explains how to measure and analyze subjective concepts like satisfaction with questionnaires. Sect. 3.5 (Statistical Evaluation), finally, explains how to statistically test the formulated hypotheses. The guidelines are illustrated with existing user-centric work in the recommender systems field where possible.

3.1 Research Model

The goal of a user experiment is to test the effect of some Objective System Aspect (OSA) on the user's Experience (EXP) and Interaction (INT). The Knijnenburg et

al. [67] framework suggests that such effects are mediated by Subjective System Aspects (SSAs), and possibly moderated by Personal and Situational Characteristics (PCs and SCs). Before conducting the experiment, the specific constructs and their expected interrelations should be presented as a *research model* consisting of a set of testable hypotheses. Each hypothesis consists of an independent variable and a dependent variable. Hypotheses are predictions about how the independent variable influences the dependent variable (and optionally, how a moderating variable qualifies this effect).

3.1.1 Determining Which OSAs Will Be Tested

The first step in developing a research model is to determine which OSAs will be tested. In a typical experiment the OSAs are manipulated independent variables (see Sect. 3.3): their presence, operation or appearance is altered between different experimental conditions, but these conditions are exactly the same otherwise (similar to A/B testing). This concept of ceteris paribus ("all else remains the same") is important, because it allows the researchers to trace differences in outcomes between conditions back to the manipulated OSA. If aside from the manipulated OSA other aspects differ between conditions as well, then these aspects are said to be *confounded* with the OSA: it is then impossible to determine whether the OSA or any of these other aspects caused the difference in outcomes.

For example, in [68] we manipulated the algorithm by testing a system with an SVD algorithm against the same system that was altered to select random items as recommendations. The items were labeled as "recommendations" in both conditions. If we had given the items different labels in each condition (e.g. "random items" and "recommendations"), then the labeling would have been confounded with the algorithm itself. I.e., if users judged the recommendations to have a higher quality, this could be either because they indeed had a higher quality, or because the "recommendations" label simply made users *think* that they had a higher quality. By having the same label for the random items, we ruled out the latter explanation.

3.1.2 Selecting Appropriate Outcome Measures (INT and EXP)

The second step in developing a research model is to select appropriate outcome measures (dependent variables). These are typically a combination of observed behaviors (INT) and questionnaire-based feedback (EXP). Although industry executives are typically most interested in objective outcomes that influence conversion rates (i.e. INT), there are reasons why the inclusion of EXP variables is beneficial for industry and academic researchers alike. First of all, users' behavior is often influenced by external factors (e.g. purchases may be gifts rather than a reflection of the user's taste; time on a page may be influenced by their Internet connection speed), so the effects of OSAs on INT are less robust than on EXP. More importantly, studies that test behavioral variables only (i.e. conventional A/B tests) can

detect behavioral differences, but they often say very little about *how and why* the behavioral difference occurred. The explanation of behavioral effects is what drives scientific discovery and sound corporate decisions, and a carefully selected combination of EXP and INT variables can provide such explanations.

Knijnenburg et al. [68] provides a good example of the importance of including both EXP and INT variables in an experiment. Looking only at the behavioral outcomes of this study, one would come to the conclusion that the system with the SVD algorithm resulted in a shorter total viewing time and fewer clips clicked than the system with random recommendations. This result may be counterintuitive, until one includes perceived system effectiveness as a mediating EXP variable: The system with the SVD recommender is perceived as more effective, which manifests in less need for browsing, and hence a shorter viewing time and fewer clips clicked. Only after incorporating both EXP and INT variables were we able to explain that the SVD recommender system is indeed effective.

Experiments that measure EXP variables require that the researchers administer questionnaires, which limits the scale of such experiments compared to conventional A/B tests. As such, A/B tests can more effectively test the behavioral effects of a large number of OSAs simultaneously (these tests are more appropriately called "multivariate tests"). The optimal test plan therefore involves both: A/B tests are used to discover interesting effects, while user experiments with questionnaires can follow up these tests to explain how and why these interesting effects come about.

Generally speaking, a well-rounded research effort should use a combination of INT and EXP variables: the EXP variables explain differences in participants' behavior, while the INT variables "ground" the user experience in observable behavior.

3.1.3 Explaining The Effects With Theory And Mediating Variables (SSAs)

The inclusion of EXP variables alone is not always sufficient to explain how and why users are more satisfied or behave differently between conditions. Moreover, even if one can demonstrate that a certain OSA makes users more (or less) satisfied, there needs to be a compelling argument about whether this finding is generalizable, or rather just a one-off event. A *theory* that explains the hypothesized effects of a study more thoroughly can provide a sense of its generalizability [45]. In this regard, reseachers can consult existing theories of user experience [46, 47], technology acceptance [26, 116], attitudes and behaviors [2, 3, 4, 37], or the theory of how users experience technology embedded in the Knijnenburg et al. [67] framework.

Just having a theory for the hypothesized effects is not enough, though; the experiment can (and should) confirm these theories. In the words of Iivari [53], this means translating the conceptual level theories to the descriptive level, which involves not only developing hypotheses regarding expected effects of the OSA on INT and EXP variables, but also hypotheses that explain *how and why* these effects come about.

A theory can also help in fine-tuning experimental conditions to rule out alternative explanations. For example, choice overload theory suggests that choice over-

load is moderated by the *diversity* of an item set, independent of its *quality* and *size* [34, 103]. In Willemsen et al. [120, 121] we therefore took care to increase the diversity of the recommendations witout reducing their quality, and we manipulated the size of the item set independently from the diversity.

Another way to test theoretical explanations is to include mediating SSA variables in the research model. These SSAs serve both as a dependent variable (in the hypothesized effect of OSA $\rightarrow$ SSA) and an independent variable (in the hypothesized effect of SSA $\rightarrow$ EXP). For example, experiment FT4 in [67] tested two matrix factorization algorithms, one using explicit feedback (MF-E) and the other using implicit feedback (MF-I), against a system that recommended the (non-personalized) most popular items. The results ([67], Fig. 9) showed that both algorithms (OSAs) result in a more effective system (EXP) than the non-personalized version, but that the reason for this differs per algorithm. Specifically, the MF-I recommendations are perceived to have a higher quality (OSA $\rightarrow$ SSA), and these higher quality recommendations eventually result in a more effective system (SSA $\rightarrow$ EXP). On the other hand, the MF-E recommendations are perceived to be more diverse (OSA $\rightarrow$ SSA), and these diverse recommendations are perceived to have a higher quality (SSA $\rightarrow$ SSA) and thus result in a more effective system (SSA $\rightarrow$ EXP). The mediating SSAs explain the different reasons why each algorithm leads to a more effective system.

Finally, it may happen that the outcome variable does not differ between OSA conditions. In some cases, a theoretical examination may point out that different underlying effects could be counteracting each other, effectively cancelling out the total effect of the OSA. One can then demonstrate this theoretical phenomenon by measuring these underlying causes and including them as mediating variables in the research model.

For example, in Bollen et al. [12] we showed that there was no effect of the experimental conditions on overall choice satisfaction, but we were still able to demonstrate the phenomenon of "choice overload" by incorporating the mediating variables item set attractiveness and choice difficulty. Specifically, the results showed that more attractive item sets led to higher choice satisfaction, but that attractive sets were also more difficult to choose from, which in turn reduced choice satisfaction. We thereby demonstrated that good recommendations do not always lead to higher choice satisfaction due to choice overload. Similarly, Nguyen et al. [87] showed that the increased effectiveness of rating support by means of providing exemplars was limited, because it was counteracted by increased difficulty of using this type of support, compared to a baseline rating scale.

3.1.4 Include PCs and SCs Where Appropriate

The final step in developing a research model is to determine which PCs and SCs may influence the outcome variable. Incorporating these aspects into the experiment will increase the robustness of the results, so they should be considered even though they are typically beyond the influence of the system.

In some cases, the effect of the OSA on the outcome variable is hypothesized not to hold universally, but only for a specific type of user or in a specific situation. In that case, this PC or SC is said to *moderate* the effect of the OSA on the outcome. Measuring the PC or SC is then crucial to determine the true effect of the OSA.

For example, in [66] we argued that domain novices and experts use different strategies to make decisions, and that their ideal recommender system would therefore require different preference elicitation methods. Our results demonstrated that novices were indeed more satisfied with a case-based preference elicitation method, while experts were more satisfied with an attribute-based preference elicitation method.

3.1.5 Practical Tip: Never Formulate a "No Effect" Hypothesis

It is important to note that with every hypothesis comes a *null hypothesis*, which argues the absence of the effect described in the hypothesis. For example:

H_0**:** There is no difference in perceived recommendation quality between algorithm A and algorithm B.

H_1**:** Participants perceive the recommendation quality of algorithm A to be higher than algorithm B.

It is common practice in scientific writing to only state H_1 and leave the null hypothesis implicit. Statistical evaluations can never directly "prove" H_1, but they can support it by rejecting H_0 [38]. Importantly though, the absence of support for H_1 does not mean that H_0 is supported instead. In other words, if the aforementioned H_1 is not supported, one cannot claim that there is no difference in perceived recommendation quality between algorithm A and B, only that the current study did not find such an effect. In fact, providing support for the absence of an effect is very difficult to do statistically [11]. Researchers are therefore advised to never formulate a "no effect" hypothesis. Experiments should always be set up in such a way that differences (not equalities) between experimental conditions prove the underlying theory.

3.2 Participants

Finding participants to take part in the experiment is arguably the most time-consuming aspect of conducting a user experiment. Participant recruitment involves a tradeoff between gathering a large enough sample for statistical evaluation, and gathering a sample that accurately reflects the characteristics of the target population. Both considerations are discussed below.

3.2.1 Sampling Participants

Ideally, the sample of participants in the experiment should be an unbiased (random) sample of the target population. Creating a truly unbiased sample is practically impossible, but if one aspires to extrapolate the study results to real-world situations, then the participants should resemble the users (or potential users) of the tested system as closely as possible.

To avoid "sampling bias", certain practices should be avoided. For example, it is very tempting to ask colleagues, students or friends to participate, but these people will arguably have more knowledge of the field of study than an average user. They may even know what the experiment is about, which may unconsciously cause them to behave more predictably. Your colleagues and friends may also be more excited about the experiment, and they may want to please you, which may lead to socially desirable answers [91, 107]. It is better when participants are "blind", i.e. when they have no "special" connection to the researcher, the system, or the experiment.

Another practice to avoid is to post a link to the study to one's Facebook or Twitter account, and ask for reposts/retweets. Again, the first-degree participants will have a connection with the researcher, and should therefore be discarded. Participants who responded to the reposts/retweets will be more likely to resemble "blind" users, but extra checks should be performed on them since they are recruited via a "snowball sampling method" [32, 49, 78, 101].

Participant recruitment messages should be phrased carefully, because their framing may influence who participates in the study and how participants approach the tested system. It is generally better to give a generic description of the study to avoid bias. Specifically, the description should focus on the task ("Test this music recommender and answer a questionnaire") rather than the purpose of the study ("We are studying users' privacy perceptions of a recommender system"). Avoid technical terms, otherwise non-expert users may feel they are not knowledgeable enough to participate (note that even the term "recommender system" itself may not be common parlance for some potential users). Also make sure that the experiment works in all major browsers (even older versions) and on both laptops and tablets.

In some cases it makes sense to limit participation in the experiment to a specific subset of users, especially when some users cannot be given a meaningful experience. For example, in [62] we tested the inspectability and control of social recommenders using TasteWeights, a music recommender that uses overlap between Facebook users' music likes and their friends' music likes to calculate recommendations. We limited participation in this experiment to Facebook users with sufficient overlap between their own music likes and those of their friends. Users with insufficiently overlapping profiles were asked to either add more music likes or leave the study. We argued that this was admissible because a real system would likely do something similar. At the same time though, this meant that our conclusions would only hold for eligible users, and not for the population at large.

3.2.2 Determining the Sample Size

User experiments need a reasonable sample size (often reported as *N*) to allow robust statistical evaluation of the hypotheses. Increasing the number of participants increases the statistical power of the experiment. Statistical power is the likelihood of detecting an effect of certain size in a sample, given that the effect indeed exists in the population. To determine the required sample size, researchers should perform a *power analysis* [22, 35] using an estimate (based on previous work) of the expected effect size of the hypothesized effects and an adequate power level (usually 85%). In recommender systems research manipulations typically have small effects (causing differences of about 0.2–0.3 standard deviations in the dependent variables) and occasionally medium-sized effects (differences of around 0.5 standard deviations). To detect a small effect (0.3 *SD*) with a power of 85% in a between-subjects experiment, 201 participants are needed *per experimental condition*. To detect a medium-sized effect (0.5 *SD*), 73 participants are needed per condition. Within-subjects experiments need far fewer participants: 102 to detect small effects, and 38 to test medium-sized effects. Note, though, that there are additional sample size requirements for advanced statistical procedures like Factor Analysis (see Sect. 3.4.2) and Structural Equation Modeling (see Sect. 3.5.3).

The results of "underpowered" studies should be mistrusted, even if they are statistically significant. Due to low power, it is very likely that the experimenters simply "got lucky" and found a spurious effect [88]. And even if the reported effects are real, the effect sizes are inevitably overstated. Moreover, a low *N* means that the study may not have an inductive base that is wide enough to generalize the findings to the entire population, because small samples are likely to be biased.

For example, one of the first user-centric evaluations of a recommender system, conducted by Sinha and Swearingen [104], employs only 19 participants. Even though the authors find some significant results, the study is severely underpowered so the conclusions cannot be generalized beyond this specific sample: the large effect sizes reported are likely to be much smaller (if not absent) in the population.

3.2.3 Practical Tip: Run Your Studies on a Crowd-Sourcing Platform

In the past, participants were often recruited through volunteer panels painstakingly built by universities, or through expensive consumer research panels managed by marketing firms. This has changed with the rise of classified advertisements and crowd-sourcing websites such as Craigslist and Amazon Mechanical Turk. Craigslist allows researchers to post user experiments in various cities under Jobs > Etcetera, and is very convenient for creating a geographically balanced sample. Amazon Mechanical Turk[6] is often used for very small tasks, but Turk workers appreciate more elaborate survey studies. A benefit of Mechanical Turk is that it has anonymous payment facilities. Requesters can set certain criteria for workers

[6] Mechanical Turk is currently only available for researchers in the United States, but various alternatives for non-US researchers exist.

that are allowed to participate, and experience has shown that it is good practice to restrict participants to U.S. workers with a high reputation [58, 92].

In our experience, the demographics of Craigslist and Mechanical Turk participants reflect the general Internet population, with Craigslist users being a bit higher educated and more wealthy. Turk workers are less likely to complain about tedious study procedures, but are also more likely to cheat [30]. Ample attention and quality checks can prevent cheaters from affecting the results. It is good practice to include a contact email address as well as an open feedback item in the study to catch unexpected problems with the experiment.

3.3 Experimental Manipulations

In a typical user experiment, one or more OSAs are manipulated into two or more experimental conditions following the ceteris paribus principle (see Sect. 3.1). OSAs can be manipulated in various ways. One can turn the OSA on or off (e.g. display predicted ratings or not), test different versions of the OSA (e.g. implicit versus explicit preference elicitation), or test several levels of the OSA (e.g. display 5, 10 or 20 recommendations). This section explains how to create meaningful experimental conditions, and how to randomly assign participants to them.

3.3.1 Selecting Conditions to Test

The goal of many user experiments is to demonstrate the superiority of some new invention: a new algorithm, preference elicitation method, or recommendation display technique. In such experiments, the condition with the new invention (called the *treatment* condition) should be tested against a reasonable *baseline* condition. A baseline should be included even when several treatment conditions are compared against each other, because the baseline condition links the study conditions to the status quo in recommender systems research.

Selecting a baseline can be difficult. For example, one could compare a recommender system against a non-personalized system, but the results of such an unbalanced comparison are usually unsurprising [114]. On the other hand, recommender systems are definitely not always better than their non-personalized variant, so a comparison with a non-personalized system may very well be justified when testing a recommender in a new domain [21]. Another option is to test against the state-of-the-art (e.g. what has proven to be the best algorithm, preference elicitation method, or recommendation display technique in previous work).

Not all manipulations consist of a specific baseline and treatment condition. Sometimes (especially when the experiment focused on the users' interaction with the recommender system rather than some new invention) there is no accepted baseline. A range of plausible conditions should then be considered in a way that maximizes the opportunity for the effect to occur, while staying within the realm of

plausibility. For example, testing a recommendation list length of 5 versus 300 recommendations is likely to produce a choice overload effect, but finding choice overload in lists of more plausible lengths (e.g. 20 items) is practically much more useful. Making the manipulation too subtle (e.g. testing lists of 5 versus 6 items) may not produce a choice overload effect, or the effect may be so small that many more participants are needed to detect it.

3.3.2 Including Multiple Manipulations

The simplest user experiment includes a single manipulation with two experimental conditions. One can also create multiple experimental conditions per manipulation, e.g. when manipulating recommendation list length one can test lengths of 5, 10 and 20. It is also possible to manipulate multiple OSAs in a single experiment, and this is especially interesting when these OSAs are expected to have an *interaction effect* on the outcome variables. Interaction effects occur when a certain manipulation has an effect in certain condition(s) of the other manipulation, but no effect (or the opposite effect) in the other condition(s) of the other manipulation.

For example, in [120] we showed that high-diversity recommendations were perceived as more attractive, were easier to choose from, and led to higher system satisfaction than low-diversity recommendations, but only for short recommendation lists (5 recommendations). In longer lists, there was no difference between high- and low-diversity recommendations. We concluded that giving users recommendation lists that are both short *and* diverse could reduce choice overload.

When multiple OSAs are considered simultaneously like in the example above, these OSAs should be manipulated independently, or *orthogonally* by creating an instance of the system for each possible combination of conditions. The example above considered a 2-by-3 experiment (2 levels of diversity, 3 list lengths), which resulted in 6 experimental conditions.

3.3.3 Setting Up Between-Subjects or Within-Subjects Randomization

There are essentially three ways in which participants can be assigned to experimental conditions. In a *between-subjects* experiment, participants are randomly assigned to one of the experimental conditions. A benefit of between-subjects experiments is that the manipulation remains hidden from the participant, since each participant sees only one condition. This also makes the experiment more realistic, because users of real systems usually also only see a single version of the system. The averages of outcome variables are compared between conditions to see if the OSA had an effect on the outcomes. By assigning participants to conditions randomly, any differences between participants are leveled out. These differences can still cause random fluctuations in the outcomes, though, which is why between-subjects experiments typically need a larger N to attain an adequate level of statistical power.

Our study on different interfaces for an energy-saving recommender [65] is a good example of a between-subjects experiment. In the experiment different preference elicitation methods are tested, and users' satisfaction with the chosen energy-saving measures is an important outcome variable in the experiment. Having participants go through the same process of choosing energy-saving measures several times would have been rather weird, and users would have been able to guess the purpose of the different preference elicitation methods, which could have affected the results. With 5 conditions and a number of moderating PCs, the 147 participants recruited for this study were a bare minimum, though.

In a *sequential within-subjects* experiment, participants interact with both experimental conditions, one at a time. A benefit of within-subjects experiments is that differences in outcomes can be compared for each participant, which effectively eliminates the between-participant variability. As a result, fewer participants are needed to attain an adequate level of statistical power. A downside is that participants may be able to guess the experimental manipulation, and that repeating the same experiment several times may feel unnatural. Moreover, participants may react differently the second time they walk through the experiment. Randomizing the order in which participants see the conditions prevents the order from becoming confounded with the condition in the overall analysis.

In [121] we provide a good example of a within-subjects manipulation. In that study we tested three levels of diversification of the recommendations. The three different recommendation lists were presented in random order. Other than containing different items, the lists showed no apparent differences, so it was not possible for participants to guess the purpose of the study. Moreover, the presented lists were sufficiently different that the task of selecting an item from the list did not feel repetitive. Due to the within-subjects setup, the study was able to detect subtle differences between conditions. The study additionally manipulated the list length between-subjects, but no differences between length conditions (or interactions with diversification) were found.

Pu and Chen [94] also use a within-subjects manipulation, to test two different presentation techniques for recommendations. Each participant completes two tasks, one with each presentation technique. To avoid repetitiveness, the tasks involve different recommendation domains (digital cameras and notebooks). The presentation order of domains and techniques are manipulated between-subjects in a 2-by-2 setup; this cancels out any order- and task-effects. They then compare the presentation techniques using within-subjects tests.

In a *simultaneous within-subjects* experiment, participants experience all conditions at the same time. This allows participants to compare the different conditions and choose which one they like best. This again reduces between-participant variability, and also avoids order effects. Note though that the position of experimental conditions should be randomized, because we do not want to confound condition with position on the screen. The advantage of this method is that it can detect very subtle differences between conditions. The downside is that showing two conditions simultaneously is obviously a far cry from a realistic usage scenario.

As an example of a simultaneous within-subjects experiment, Ghose et al. [43] considered a novel ranking algorithm for a hotel and travel search site based on crowd-sourced content. Their study pairs the proposed algorithm with several different baseline algorithms. Each pair is tested as a simultaneous within-subjects experiment, where the two rankings produced by the proposed algorithm and the baseline algorithm are presented side-by-side, and users choose which ranking they prefer. The results show that their proposed algorithm is significantly preferred over 13 different baselines in six different cities. On average, twice as many participants prefer the recommendations of the proposed algorithm to the baseline.

Ekstrand et al. [31] also conducted a simultaneous within-subject design, and they chose this design because they were interested in detecting subtle differences between two recommendation lists produced by common algorithms (user-user, item-item and SVD). Like Ghose et al. [43] Users were asked which list they preferred, but also to indicate perceived *differences between* the lists in terms of the relative satisfaction, novelty and diversity. Importantly, Ekstrand et al. were able to link these perceived differences to objective measures of recommendation quality (e.g., perceived novelty was predicted by popularity rank). The results show that novelty (which was highest for the user-user algorithm) had a negative effect on satisfaction and preference for a list, whereas diversity showed a positive effect.

Increased realism is the main reason why between-subjects experiments are more appropriate than within-subjects experiments in most recommender system studies. Note, however, that even a between-subjects experiment is not completely natural: participants know that they are part of an experiment, and may therefore behave differently. This is called the *Hawthorne effect* [75]. In experiments that involve real systems, the Hawthorne effect can be detected by comparing the behavior of participants in (the baseline condition of) the experiment with the behavior of participants in the real system (or in an A/B test). If behaviors are substantially different, this is likely due to the Hawthorne effect.

3.3.4 Practical Tip: Think Big, Start Small

Designing experimental manipulations often involves difficult trade-offs. With several orthogonal manipulations with multiple variants each, the number of experimental conditions will grow exponentially. Since the number of participants needed to attain a certain level of statistical power grows linearly with the number of conditions, it is advisable to keep the number of conditions low.

The best strategy is therefore to think big, but start small: write down all possible versions of all OSAs that are relevant to the study in an experiment plan, but then start investigating the manipulation that seems most likely to cause an effect. If this experiment indeed detects the effect, subsequent experiments can be conducted to test different levels of the manipulation, or to include additional manipulations that may moderate (i.e. interact with) the existing effect.

In [16], for example, Chen and Pu identified several OSAs that may influence the effectiveness and usability of critiquing-based recommender systems: the number of

recommendations presented in the first round of preference elicitation, the number of alternatives presented after each round of critiquing, and whether the user initiates the critiquing or the system suggests critiques (for both unit critiques and compound critiques). They systematically explored these parameters in a series of 2-condition experiments. By keeping the setup of the experiments consistent, they were even able to make comparisons across experiments.

Consistent with the "think big, start small" mantra, it is in some cases perfectly acceptable to simplify a system to increase experimental control. For example, the original TasteWeights system [14] allows you to inspect connections between liked items, friends, and recommendations, and control the weights of both liked items and friends. In our user experiment of this system [62] we wanted to test the influence of these features separately, so we split the interaction into two steps: a control step and an inspection step. This allowed us to manipulate the control and inspection OSAs independently, which resulted in a much "cleaner" experimental design.

3.4 Measurement

In this section we present best practices for measuring perceptions (SSAs), experiences (EXPs) and personal and situational characteristics (PCs and SCs) using questionnaires. Most importantly, we give the reader a practical example of performing a Confirmatory Factor Analysis (CFA) using MPlus[7], a state-of-the-art statistical software package, and Lavaan[8] a package for R that has many of the same features.

3.4.1 Creating Measurement Scales

Due to their subjective nature, measuring perceptions, experiences, and personal and situational characteristics is not as easy as it may seem. Whereas objective traits can usually be measured with a single question (e.g. age, income), this is not advisable for subjective concepts. Single-item measurements such as "On a scale from 1 to 5, how much did you like this system?" are said to lack *content validity*: each participant may interpret the item differently. For example, some may like the system because of its convenience, others may like it because of its ease of use, and again others may like it because the recommendations are accurate. These different interpretations reduce the precision and conceptual clarity of the measurement.

A better approach is to create measurement scales consisting of multiple items;[9] at least 3 but preferably 5 or more. This is a delicate process that usually involves multiple iterations of testing and revising items. It is advisable to first develop around 10–15 items and then reduce it to 5–7 through discussions with domain

[7] http://www.statmodel.com/

[8] http://lavaan.ugent.be/

[9] Or, multiple measurement scales for the different constructs (e.g. system satisfaction, ease of use, and recommendation quality), each measured with mutiple items.

experts and comprehension pre-tests with test subjects. 1–2 additional items may still be discarded during the analysis of the actual study results.

The items in most user experiments are phrased as statements (e.g. "The system was easy to use") to which participants are asked to express their agreement on a 5- or 7-point scale (from "strongly disagree" to "strongly agree"). Studies have shown that participants find such items easy to answer. There are a few additional tips for designing good questionnaire items:

- Invest a lot of time in deciding upon a clear definition of the construct to be measured, and check for each item whether it fits the construct definition.
- Include both positively and negatively phrased items. This will make questionnaires less leading, and allows one to explore the flipside of the construct. It also helps to filter out participants who do not carefully read the items. However, avoid the word "not", because it is too easily overlooked.
- Study participants may not have a college degree, so their reading level may be low. Use simple words and short sentences to aid comprehension. Like with the recruitment message, try to avoid technical terms.
- Avoid double-barreled questions. Each item should measure only one thing at a time. For example, if a participant found the system fun but not very useful, they would find it hard to answer the question "The system was useful and fun."

As mentioned, it is a good idea to pre-test the questionnaire items with experts; they can give advice on how to accurately define the concept to be measured, and on whether the proposed questionnaire items cover all aspects of the concept. Furthermore, comprehension pre-tests can be conducted to test how well participants understand the questionnaire items. A comprehension pre-test invites participants to read the questionnaire items aloud and to explain their reasoning while answering the questions. Their think-aloud answers can highlight questionaire items that are unclear or interpreted incorrectly.

3.4.2 Establishing Construct Validity

Once a set of items has been developed that accurately reflects the concept to be measured (i.e. content validity is established), the next step is to establish *construct validity*, i.e. to make sure that the items comprise a robust and valid measurement scale. For the purpose of statistical analysis, each multi-item measurement scale has to be turned into single variable. Summing the item scores may seem like the most straightforward way of doing this, but Confirmatory Factor Analysis (CFA) is a more sophisticated solution that not only creates the measurement variable but also tests some of the preconditions for construct validity along the way.

Listings 1 and 2 show example input of a CFA as ran in MPlus and Lavaan. The output of these tools is very similar, so we present it for MPlus only (Listing 3). The example CFA is based on an experiment with a social network based music recommender system [62]. This system employs an innovative graph-based interface that shows how the users' Facebook music "likes" overlap with their friends' music

"likes", and how these friends' other music "likes" are in turn used to create a set of recommendations. In the graph, users can trace back each recommendation to the friends that "liked" that item, and to the overlapping "likes" that caused these friends to be part of the user's nearest-neighborhood. We argued that this graph would provide a good justification for the recommendations, thereby increasing the perceived recommendation quality (`quality`) and the understandability of the recommender system (`underst`). Moreover, we allowed users to control either the weights of their "likes" or the weights of their friends, and we argued that this would influence their perceived control (`control`). Finally, we argued that perceived recommendation quality, understandability, and control would ultimately increase users' satisfaction with the system (`satisf`).

The CFA validates the four subjective measurement scales of the experiment. Each scale is represented by a latent factor, with each item loading on its designated scale (MPlus: lines 8–11, Lavaan: lines 2–5). The output shows the loadings of the items on the factors (lines 1–30), which are proportional to the extracted variance (lines 42–67). The factors may be correlated with each other (lines 32–40). The solution has no standard scale, so we include code (MPlus: line 12, Lavaan: lines 6–9) to give the factors a standard deviation of 1 and a mean of 0.[10] We also declare all items as ordered categorical (MPlus: line 6, Lavaan: line 12), because they are measured on a 5-point scale. Otherwise, the items would be treated an interval scale, which would assume that the difference between "completely disagree" (1) and "somewhat disagree" (2) is the same as the difference between "neutral" (3) and "somewhat agree" (4). MPlus and Lavaan model ordered categorical variables in a way that does not make this assumption.

Listing 1 CFA input, MPlus

```
1  DATA: FILE IS twc.dat;   !specify the data file
2  VARIABLE:   !list the variable names (columns in the data file)
3   names are s1 s2 s3 s4 s5 s6 s7 q1 q2 q3 q4 q5 q6
4    c1 c2 c3 c4 c5 u1 u2 u3 u4 u5 cgraph citem cfriend;
5   usevariables are  s1-u5;   !specify which vars are used
6   categorical are s1-u5;   !specify which vars are categorical
7  MODEL:   !specify each factor as [factorname] by [vars]
8   satisf  by s1* s2-s7;   !satisfaction
9   quality by q1* q2-q6;   !perceived recommendation quality
10  control by c1* c2-c5;   !perceived control
11  underst by u1* u2-u5;   !understandability
12  satisf-underst@1;   !set the std. dev. of each factor to 1
```

[10] MPlus and Lavaan use a different parameterization by default by fixing the loading of the first item to 1. We free up these loadings by including an asterisk after (MPlus) or `NA*` before (Lavaan) the first item of each factor. This alternative solution conveniently standardizes the factor scores.

Listing 2 CFA input, Lavaan (R package)

```
model <- '   #specify each factor as [factorname] =~ [vars]
 satisf  =~ NA*s1+s2+s3+s4+s5+s6+s7   #satisfaction
 quality =~ NA*q1+q2+q3+q4+q5+q6   #perceived rec. quality
 control =~ NA*c1+c2+c3+c4+c5   #perceived control
 underst =~ NA*u1+u2+u3+u4+u5   #understandability
 satisf  ~~ 1*satisf   #set the std. dev. of each factor to 1
 quality ~~ 1*quality
 control ~~ 1*control
 underst ~~ 1*underst
';
fit <- sem(model, data=twc,   #specify the dataset
 ordered=names(twc));   #specify which vars are categorical
summary(fit, rsquare=TRUE);   #produce model fit and R^2 values
```

Listing 3 CFA output

```
MODEL RESULTS

                                                    Two-Taile
                   Estimate       S.E.  Est./S.E.    P-Value
 SATISF   BY
    S1                0.887      0.018     49.604      0.000
    S2               -0.885      0.018    -48.935      0.000
    S3                0.770      0.029     26.982      0.000
    S4                0.821      0.025     32.450      0.000
    S5                0.889      0.018     50.685      0.000
    S6                0.788      0.031     25.496      0.000
    S7               -0.845      0.022    -38.426      0.000
 QUALITY  BY
    Q1                0.950      0.013     72.837      0.000
    Q2                0.949      0.013     73.153      0.000
    Q3                0.942      0.012     77.784      0.000
    Q4                0.805      0.033     24.332      0.000
    Q5               -0.699      0.042    -16.700      0.000
    Q6               -0.774      0.040    -19.428      0.000
 CONTROL  BY
    C1                0.711      0.038     18.653      0.000
    C2                0.855      0.024     35.667      0.000
    C3                0.906      0.022     41.704      0.000
    C4                0.722      0.037     19.276      0.000
    C5               -0.425      0.056     -7.598      0.000
 UNDERST  BY
    U1               -0.568      0.048    -11.745      0.000
    U2                0.879      0.019     46.539      0.000
    U3                0.748      0.031     24.023      0.000
    U4               -0.911      0.020    -46.581      0.000
    U5                0.995      0.014     70.251      0.000
 QUALITY  WITH
    SATISF            0.686      0.033     20.541      0.000
 CONTROL  WITH
    SATISF           -0.760      0.028    -26.962      0.000
    QUALITY          -0.648      0.040    -16.073      0.000
 UNDERST  WITH
```

```
37      SATISF             0.373      0.049      7.581      0.000
38      QUALITY            0.292      0.059      4.932      0.000
39      CONTROL           -0.396      0.051     -7.736      0.000
40
41  R-SQUARE
42      Observed                     Residual
43      Variable        Estimate     Variance
44      S1                 0.788        0.212
45      S2                 0.783        0.217
46      S3                 0.593        0.407
47      S4                 0.674        0.326
48      S5                 0.790        0.210
49      S6                 0.622        0.378
50      S7                 0.714        0.286
51      Q1                 0.903        0.097
52      Q2                 0.901        0.099
53      Q3                 0.888        0.112
54      Q4                 0.648        0.352
55      Q5                 0.488        0.512
56      Q6                 0.599        0.401
57      C1                 0.506        0.494
58      C2                 0.731        0.269
59      C3                 0.820        0.180
60      C4                 0.521        0.479
61      C5                 0.180        0.820
62      U1                 0.322        0.678
63      U2                 0.772        0.228
64      U3                 0.560        0.440
65      U4                 0.831        0.169
66      U5                 0.990        0.010
```

As mentioned earlier, an advantage of using CFA over simply summing the item scores is that it can help establish the construct validity of the measurement scales. Specifically, CFA can be used to establish convergent and discriminant validity. Convergent validity determines whether the items of a scale measure a single construct (i.e. that the scale is not a combination of multiple constructs, or simply a collection of items with no common ground), while discriminant validity determines whether two scales indeed measure two separate constructs (i.e. that two scales are not so similar that they actually measure the same construct).

Convergent validity is said to hold when the average variance extracted (AVE) from the items measuring the factor is larger than 0.50. Beyond that, a higher AVE indicates more precise measurement. The AVE can be calculated by averaging the R^2 values for all items of a factor (e.g., lines 54–60 for `satisf` and lines 61–66 for `quality`). The AVE can be improved by iteratively removing items with low loadings. Doing this for the presented data removes items `C5`, `U1` and `U3` from the model, respectively. Bear in mind that at least three items should remain per factor, because a factor with only two items has no free parameters for estimation. Generally speaking, more items provide a better definition of the construct, and aiming for 4–5 items per construct is good practice.

In some cases convergent validity does not hold because a factor actually measures more than one construct. For example, in [63] we found that information disclosure to an app recommender system actually consisted to two correlated factors: demographics disclosure and context data disclosure. If there exists some uncertainty about the factor structure, an Exploratory Factor Analysis (EFA) can be used to discover the correct factor structure.[11] EFA initially makes no assumptions about which items load on which factors, but tries to find a "clean" factor structure (with each item loading on one of the factors) that best fits the data. In [64] we employ this technique to discover the various dimensions of information disclosure in three different datasets. We first run several EFAs with an increasing number of factors to determine the optimal number of dimensions (looking at fit statistics and the conciseness of the model). Then we inspect the model to determine the optimal factor structure, and conduct a CFA to generate the final measurement model.

Discriminant validity is called into question when two scales are too highly correlated (i.e. when the correlation is higher than the square root of the AVE of either of the two factors). In that case the scales measure essentially the same thing, which means that they can be combined, or that one of the scales can be discarded. For example, in FT2 of [67] we originally tried to measure separate factors for perceived usefulness and fun. These factors were however so highly correlated that we ended up integrating them into a single factor.

There is no consensus on the sample size needed for CFA, but 100 participants seems to be a bare minimum, or 200 when unvalidated factors are tested [79]. Larger CFAs probably require even more participants: a rule of thumb is to have at least 5 participants per questionnaire item.

3.4.3 Practical Tip: Use Existing Scales

Developing measurement scales from scratch is a time-consuming activity. Researching new phenomena often calls for specialized measurement scales, so this effort is in many cases unavoidable. A good tip is to look for related measurement scales and adapt them to the experiment at hand. For example, in [70] we developed scales for privacy concerns and protection as system- and provider-specific versions of existing scales. Surprisingly little scale development work has been done in the Human-Computer Interaction field; the Management Information Systems field is a much better source for related scales.

Most experiments also include some more general constructs that can be copied verbatim from existing work (this is considered good practice, not plagiarism). Two sources for existing scales related to recommender systems are the Knijnenburg et al. [67] framework paper and the ResQue framework developed by Pu, Chen and Hu [95]. In Knijnenburg et al. [67] we include scales for the following concepts:

[11] Moreover, even if you are more or less certain about the factor structure of a CFA model, it pays to consult the *modification indices* of the model. The use of modification indices and CFA goes beyond the current chapter, but is thoroughly explained in Kline's [59] practical primer on Structural Equation Models.

- Perceived recommendation quality (SSA)
- Perceived recommendation accuracy (SSA)
- Perceived recommendation variety (SSA)
- Perceived system effectiveness (and fun) (EXP)
- Choice Difficulty (EXP)
- Choice Satisfaction (EXP)
- Effort to use the system (EXP)
- Intention to provide feedback (INT)
- General trust in technology (PC)
- System-specific privacy concern (SC)

Pu, Chen and Hu [95] include scales for the following concepts (classification ours, only scales with more than 2 items are included):

- Interface adequacy (SSA)
- Interaction adequacy (SSA)
- Control (SSA)
- Perceived usefulness (EXP)
- Confidence and trust (EXP)
- Use Intentions (INT)

Despite the fact that the measurement properties of these scales have been tested before, it is still wise to perform factor analysis on new experimental data to make sure that the constructs are robustly measured in the context of the new experiment.

3.5 Statistical Evaluation

Once the validity of measurements is established and scales have been constructed, the next step is to statistically test the formulated hypotheses. Note that the practice of statistical evaluation is continuously evolving, developing tests that are ever stronger and more robust. One of the most prominent changes is the transition from piecewise statistical testing to integrative approaches that evaluate entire research models and provide simultaneous tests of all hypothesized effects.

As most scholars have been trained in piecewise statistical testing (primarily t-tests, ANOVAs, and regressions), we will briefly discuss this approach first, but assume that the reader is already familiar with the mechanics of conducting such tests. Instead, we will focus mainly on the assumptions that such tests make about the data, and the consequences when these assumptions are violated. Subsequently we will discuss the integrative approach in more detail by giving the reader a practical example of testing a Structural Equation Model (SEM) in MPlus and Lavaan.

3.5.1 Piecewise Statistical Testing: T-tests, ANOVAs, and Regressions

Most researchers perform piecewise tests of their hypotheses, which means that they perform a separate test of each dependent variable. The dependent variable is typically a continuous variable that is either an observed behavior (INT) or a measured construct (SSA or EXP). For measured constructs, individual item scores are transformed into a scale score, either by saving the factor scores from the CFA or by simply summing the item scores (after establishing construct validity with a CFA). The independent variables can either be manipulated OSAs (i.e. the experimental conditions), continuous variables (SSA, EXP or INT), or both.

The difference between two experimental conditions (e.g., the effect of a manipulated OSA on a continuous outcome) can be tested with a t-test. For between-subject manipulations (see Sect. 3.3), one uses an independent (2-sample) t-test. For within-subjects manipulations, one should use a paired (1-sample) t-test.

The main outcome of a t-test is the t-statistic and its p-value; a smaller p-value signifies more evidence against the null-hypothesis. We typically reject the null hypothesis at $p < .05$. It is important to also look at the actual difference in the dependent variable between the experimental conditions: does this difference signify a substantial effect? For example, the difference between spending $150 and $151 in an e-commerce recommender may not be substantial enough to be practically relevant, especially if that difference is caused by a computationally expensive new recommendation algorithm.

The difference between more than two conditions can be tested with an ANOVA (or a repeated measures ANOVA in case of a within-subjects design). The ANOVA test produces an F-statistic; its p-value signifies evidence against the null hypothesis that the dependent variable has the same value in all conditions. When this "omnibus" test is significant, it is usually followed up by testing specific conditions against each other.

Multiple manipulations can be tested simultaneously with a factorial ANOVA. Factorial ANOVA tests exist for between-subjects, within-subjects and mixed (both within- and between-subjects) experiments. The factorial ANOVA will provide test statistics for each manipulation as well as the interaction between the manipulations. Due to the complexity of such interaction effects, it is often helpful to plot the mean of the dependent variable for each (combination of) experimental condition(s). Visually inspecting this plot will give you a good understanding of the effects; the ANOVA results can then be used to find out whether these effects are likely to be real or due to chance variation.

The effect of one or more continuous independent variables on a continuous dependent variable can be tested with a linear regression (or a multilevel regression in case of a within-subjects design). Each independent variable receives a β-weight, which signifies the effect of a 1-unit difference in the independent variable on the dependent variable. A t-statistic and a p-value signify the evidence against the null hypothesis that this β-weight is zero. The regression also has an R^2-value, which is the percentage of the variance of the dependent variable that is explained by the set of independent variables.

Combinations of continuous independent variables and experimental manipulations can be tested with either a linear regression or an ANCOVA; note that all the mentioned tests are essentially special cases of linear regression, so a linear regression can in principle be used in any of the mentioned situations.

3.5.2 Assumptions of Statistical Tests

The real art of statistical evaluation is to know when *not* to apply a certain statistical test. Virtually all statistical tests make certain assumptions about the data, and violating these assumptions may invalidate the results of the test.

A very common violation is that of *multiple comparisons*. The purpose of any statistical test is to decide whether an observed effect is "real" or due to chance variation. Taking $p < .05$, we essentially allow an error margin of 5%: only 1 out of every 20 chance variations is expected to test significantly. However, if we have k conditions and we test for differences between all possible pairs of conditions, the *family-wise error* (i.e. the chance that *at least one* chance variation tests significantly) grows considerably. At $k = 5$ this amounts to 10 tests, and the family-wise error rate is 40%. To prevent this problem, one should always perform an omnibus test (e.g. the F-test in ANOVA) to first make sure that there *are* differences between conditions. Next, one can pick a baseline condition and compare all conditions against that condition, or one can perform all pairwise tests but calculate a more stringent p-value using post-hoc test methods such as the *Bonferroni correction*.

Another common violation is that of *data type* and *non-normality*. The t-test, ANOVA and regression all assume that the dependent variable is a normally distributed interval[12] variable that is unbounded within its predicted range. This is by definition true for factor scores (SSA and EXP), but not for most interaction variables (INT) such as number of clicks, time (bounded by zero), star ratings (bounded and discrete), or purchase decisions (yes/no). Certain non-normality problems can be solved by applying a formulaic transformation to the dependent variable to make its distribution more normal. For example, most zero-bounded variables such as time become more normal by applying a log transformation: $x_t = \ln(x + a)$, where a is a fraction of x, chosen in such a way that x_t has a fairly normal distribution. Data type problems can be accounted for by using generalized linear models (GLMs) or robust regression algorithms. For example, logistic regression can test nominal outcomes, and Poisson or negative binomial regressions can model count data. Many textbooks suggest the use of non-parametric tests, but these are old-fashioned solutions to non-normality problems, and typically do not work for non-continuous data types; GLMs and robust regressions are typically much more powerful ways to deal with non-normal data and alternative data types.

Arguably the most severe violation is that of *correlated errors*. This problem occurs when repeated measurements on the same participant are treated as indepen-

[12] An important property of the "interval" data type is that differences between values are comparable. This is for instance not true for a rating score: the difference between 1 and 2 stars is not necessarily the same as the difference between 3 and 4 stars (cf. [74]).

dent. Repeated measurements do not only occur in within-subjects experiments, but also when a certain variable is measured several times, such as the lengths of several sessions from the same participant, or the ratings of several items per session. One can solve this problem by taking the average of the repeated measurements and do the analysis with those average values, but this reduces the number of observations (and thereby the statistical power), and that it becomes impossible to make inferences about individual sessions/ratings/etc. An alternative solution is to use an advanced regression method that allows one to estimate the error correlations resulting from repeated measurements (i.e. multilevel regression).

Advanced regression techniques have been developed for data that are both non-normal and repeated, e.g. generalized linear mixed models (GLMM) and generalized estimating equations (GEE). The algorithms implementing these methods are under continuous development. Due to the complexities of such analyses, it is a good advice to consult a statistician if your data happens to have such structure.

3.5.3 Integrative Statistical Testing: Structural Equation Models

In this section we present the state-of-the-art of statistical testing: Structural Equation Modeling (SEM). SEM is an integrative statistical procedure, because it tests the measurement model and all hypotheses (known as the structural model, or path model) at the same time. Practically speaking, a SEM is a CFA where the factors are regressed on each other and on the experimental manipulations. Observed behaviors (INT) can also be incorporated in SEM.

Listings 4–6 present example input and output of a SEM as ran in MPlus and Lavaan, using the same example as the CFA ([62], see section 3.4.2), but adding the two experimental manipulations of the experiment. The 'control' manipulation has three conditions: In the 'item control' condition participants can set a weight for each their "likes", which in turn determines the weight for each friend that also likes these items. In the 'friend control' condition participants can set a weight for each of their friends directly. Finally, in the 'no control' condtion participants do not set any weights at all (i.e. items are weighted equally, and friend-weights are based on the number of overlapping items). This manipulation is represented by two dummies: `citem` is 1 for participants in the 'item control' condition; `cfriend` is 1 for participants in the 'friend control' condition. Both variables are 0 for participants in the 'no control' condition, making this the baseline condition.

The 'inspectability' manipulation has two conditions: In the the 'full graph' condition participants get to see the graph-based interface; in the 'list only' condition they get to see a list of recommendations only. This manipulation is represented by the dummy variable `cgraph`, which is 1 for participants in the 'full graph' condition and 0 for participants in the 'list only' baseline condition.[13]

[13] Here we do not discuss the interaction effect between inspectability and control. This interaction can be tested by multiplying their dummies, creating `cgraphitem` and `cgraphfriend`. These dummies represent the additional effect of item- and friend-control in the graph condition (and likewise, the additional effect of the graph in the item- and friend-control conditions).

For the CFA part of the model we specify the optimized CFA with the items `C5`, `U1` and `U3` removed (MPlus: lines 8–12, Lavaan: lines 2–9; the CFA output is excluded for brevity). The input now also includes a structural part that specifies the regressions of each dependent variable on the independent variables (MPlus: lines 13–16, Lavaan: lines 10–13). The output of these regressions (lines 18–46) can be interpreted as traditional regression outcomes with β-weights, standard errors, a test statistic, and a p-value. The β-weight for `cgraph` tests the difference between the 'full graph' and 'list only' condition, while the β-weights for `citem` and `cfriend` compare these conditions with the 'no control' condition. We conduct an omnibus test for the effect of the control manipulation on understandability (MPlus: lines 16–17, Lavaan: lines 13 and 17), and the output shows that the overall effect of this manipulation is significant (lines 6–9).

Listing 4 SEM input, MPlus

```
1  DATA: FILE IS twc.dat;
2  VARIABLE:
3   names are s1 s2 s3 s4 s5 s6 s7 q1 q2 q3 q4 q5 q6
4    c1 c2 c3 c4 c5 u1 u2 u3 u4 u5 cgraph citem cfriend;
5   usevariables are  s1-c4 u2 u4 u5 cgraph citem cfriend;
6   categorical are s1-u5;
7  MODEL:   !specify regressions as [factor] on [predictors]
8   satisf  by s1* s2-s7;
9   quality by q1* q2-q6;
10  control by c1* c2-c5;
11  underst by u1* u2-u5;
12  satisf-underst@1;
13  satisf  on  quality control underst cgraph citem cfriend;
14  quality on control underst cgraph citem cfriend;
15  control on underst cgraph citem cfriend;
16  underst on cgraph citem cfriend (p1-p3);
17 MODEL TEST: p2=0; p3=0;   !conduct the omnibus test
```

Listing 5 SEM input, Lavaan (R package)

```
1  model <- '   #specify regressions as [factor] ~ [predictors]
2   satisf  =~ NA*s1+s2+s3+s4+s5+s6+s7
3   quality =~ NA*q1+q2+q3+q4+q5+q6
4   control =~ NA*c1+c2+c3+c4+c5
5   underst =~ NA*u1+u2+u3+u4+u5
6   satisf  ~~ 1*satisf
7   quality ~~ 1*quality
8   control ~~ 1*control
9   underst ~~ 1*underst
10  satisf  ~ quality+control+underst+cgraph+citem+cfriend
11  quality ~ control+underst+cgraph+citem+cfriend
12  control ~ underst+cgraph+citem+cfriend
13  underst ~ cgraph+p2*citem+p3*cfriend
14 ';
15 fit <- sem(model, data=twc, ordered=names(twc[1:23]));
16 summary(fit, fit.measures=TRUE);
17 wald(fit, "p2;p3");   #conduct the omnibus test
```

Listing 6 SEM output

```
MODEL FIT INFORMATION
 Chi-Square Test of Model Fit
         Value                             341.770*
         Degrees of Freedom                    212
         P-Value                            0.0000
 Wald Test of Parameter Constraints
         Value                               9.333
         Degrees of Freedom                      2
         P-Value                            0.0094
 RMSEA (Root Mean Square Error Of Approximation)
         Estimate                            0.048
         90 Percent C.I.                     0.038  0.057
         Probability RMSEA <= .05            0.637
 CFI/TLI
         CFI                                 0.990
         TLI                                 0.988

MODEL RESULTS
                                                              Two-Tailed
                    Estimate       S.E.  Est./S.E.    P-Value
                        <CFA output excluded>
 SATISF   ON
    QUALITY            0.434      0.077      5.600      0.000
    CONTROL           -0.833      0.111     -7.492      0.000
    UNDERST            0.109      0.079      1.374      0.169
 QUALITY  ON
    CONTROL           -0.761      0.086     -8.827      0.000
    UNDERST            0.055      0.077      0.710      0.478
 CONTROL  ON
    UNDERST           -0.320      0.070     -4.579      0.000
 SATISF   ON
    CGRAPH             0.036      0.145      0.249      0.803
    CITEM              0.104      0.180      0.577      0.564
    CFRIEND           -0.205      0.183     -1.122      0.262
 QUALITY  ON
    CGRAPH             0.105      0.147      0.716      0.474
    CITEM              0.093      0.158      0.586      0.558
    CFRIEND            0.240      0.190      1.262      0.207
 CONTROL  ON
    CGRAPH            -0.155      0.141     -1.099      0.272
    CITEM             -0.010      0.171     -0.058      0.954
    CFRIEND           -0.116      0.165     -0.701      0.483
 UNDERST  ON
    CGRAPH             0.524      0.137      3.834      0.000
    CITEM              0.342      0.166      2.060      0.039
    CFRIEND            0.484      0.163      2.977      0.003
```

The structural part of a SEM should be specified in accordance with the study hypotheses. However, if we *only* include the hypothesized effects, one may overlook important additional effects. For example, our hypotheses may suggest that the inspectability and control manipulations increase users' understandability and per-

ceived control, that understandability and perceived control increase the perceived recommendation quality, and that this in turn increases system satisfaction. These hypotheses assert that understandability and control have a mediated (indirect) effect on system satisfaction, but it is perfectly plausible that there also be a *direct* effect. Similarly, the hypotheses assert a direct effect of understandability on perceived recommendation quality, but it is possible that this effect is actually mediated by perceived control. A prudent way to specify the structural part of a SEM is therefore to start with a "saturated" path model of the core variables of the study (i.e. OSA, SSA and EXP), and then prune any non-significant effects from this model.

To build a saturated path model, first line up the core variables in the predicted order of cause and effect. The Knijnenburg et al. [67] framework suggests a general order: OSA → SSA → EXP. If there are multiple SSA or EXP, one should try to find theoretical or empirical arguments for a certain causal direction among them. In the example, we argue `cgraph`, `citem` and `cfriend`[14] → `underst` → `control` → `quality` → `satisf`. Next, set up all possible regressions that adhere to the correct causal direction; this is the model we ran in our example. The output of the example shows that several effects in this saturated model are non-significant. The next step is to iteratively prune the model from non-significant effects until all effects are significant at $p < .05$ (or for experiments with a very large sample, $p < .01$). In our example, we would iteratively remove non-significant effects on lines 25, 28, and 31–42. This "trimmed" SEM is presented graphically in Fig. 2; this is a standardized way to present the outcomes of a SEM analysis. Finally, we add the hypothesized effects of SCs, PCs and INTs to the model. The final SEM of our example is presented graphically in Fig. 3 of [62].

The main benefit of SEM over other statistical methods is that it estimates the measured factors and all hypothesized paths in a single model. This has several advantages over a piecewise analysis. First of all, SEM explicitly models the mediated structure of causal effects. For example, Fig. 2 shows that the effect of understandability on perceived recommendation quality is *fully mediated* by perceived control.

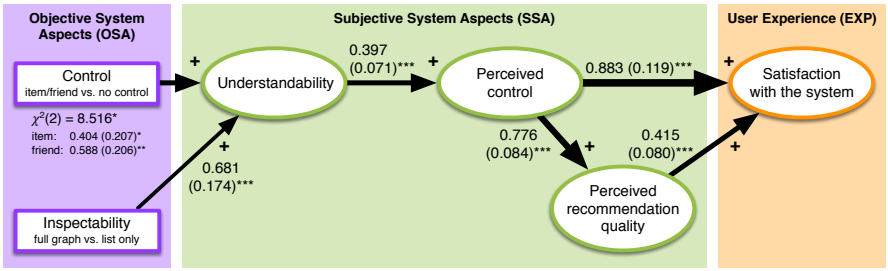

Fig. 2 The structural equation model of the trimmed SEM example. Significance levels: *** $p < .001$, ** $p < .01$, 'ns' $p > .05$. Numbers on the arrows (and their thickness) represent the β-coefficients (and standard error) of the effect. Factors are scaled to have an SD of 1.

[14] By design, experimental manipulations can only be independent variables (i.e. they never have incoming arrows), so they always start the causal chain.

In common terms: understandability leads to better recommendations because (and *only* because) understandability increases users' perceived control over the recommendations. Another example: the effect of perceived control on satisfaction is *partially mediated* by perceived recommendation quality. In common terms: control increases users' satisfaction partially because it leads to better recommendations, and partially because of other, unobserved reasons. These other reasons can be explored in a follow-up study. The ability to argue about the causal structure of a model is the main scientific advantage of SEM over piecewise statistical analyses. Mediated effects can be tested in piecewise models as well, but only in a very cumbersome, post-hoc fashion.

Secondly, in SEM the quality of the entire model itself can be evaluated with a number of fit statistics (lines 1–5 and 10–16). The Chi-square Test of Model Fit tests the difference between the predicted and observed covariance matrix. A significant test means that there is significant misfit between the model and reality. Models are an abstraction of reality, though, so a certain amount of misfit is expected, and this often amounts to significant misfit [8]. The alternative fit indices (*CFI*, *TLI*, and *RMSEA*) give an indication of how much misfit the model contains. Hu and Bentler [51] propose cut-off values for these indices to be: $CFI > .96$, $TLI > .95$, and $RMSEA < .05$ for a good model. The 90% confidence interval on the *RMSEA* indicates the precision with which the amount of misfit is predicted. This interval will be wider in smaller samples, and should remain below .10. The model fit statistics help researchers in their effort to find a well-fitting model.[15]

Finally, there is a technical advantage to fitting the measurement model and the structural model simultaneously. Psychological constructs are never measured with 100% precision, even when they are measured with multiple items. This lack of precision leads to measurement error, which attenuates the structural effects. In SEM, however, the precision of a factor can be estimated, and the structural effects can be corrected for measurement error, leading to more powerful statistical tests and thus a more robust statistical analysis. Note that despite this additional power, SEM is not a suitable method for analyzing data from small samples; estimating a reasonably complex SEM model requires data from at least 200 participants [55, 59].

3.5.4 Practical tip: Learn More About Structural Equation Modeling

MPlus and the Lavaan R package are but examples of tools to analyze Structural Equation Models. Other tools include AMOS and Lisrel, and several different R packages. We recommend the use of MPlus because it is easy to learn, has a powerful set of advanced modeling features, and it uses non-normality robust estimators by default. It also has good online support and an expansive collection of high quality video lectures covering a wide range of simple and advanced modeling techniques. We advise any reader who is serious about SEM to go to http://www.statmodel.com/ and watch these videos. Beyond these videos, Kline [59]

[15] Like in CFA, more exploratory model efforts can be assisted by the use of modification indices. Please consult [59] for examples.

provides a more general introduction to SEM, and Bollen [13] is the most comprehensive technical reference.

4 Conclusion

When we first endeavored to explain the process of conducting user experiments in [69], we presented it with the following four steps:

1. Assign participants to conditions
2. Log interaction behavior
3. Measure subjective experience
4. Analyze the collected data

Following an overview of our user-centric evaluation framework and a discussion of interesting recommender system aspects to evaluate, the practical guidelines in this chapter provide a more comprehensive discussion of the steps involved in conducting user experiments. These guidelines first emphasized the formulation of testable hypotheses. They then discussed the importance of collecting an unbiased sample of participants that is large enough to test the hypothesized effects. Next, they covered the development of distinct experimental conditions that manipulate relevant system aspects, as well as different ways of randomly assigning participants to these conditions. The guidelines then covered the practice of measuring subjective constructs that can be used to determine the perceptual and evaluative effects of the experimental manipulations. Finally, they explained in detail how to statistically evaluate the formulated hypotheses with the collected data.

By now it should be clear that learning about user experiments requires working knowledge in several related domains: It involves familiarizing oneself with the basic theory of human-computer interaction and human decision-making, research methods, psychometrics and scale development, and statistics. This chapter has touched upon each of these topics briefly, but we encourage readers to continue their learning process in each of these directions. To this effect, we include a selection of excellent textbooks and other sources below:

On human-computer interaction and human decision-making

- Jacko, "The Human-Computer Interaction Handbook: Fundamentals, Evolving Technologies, and Emerging Applications" [54]: A thorough primer on Human-Computer Interaction. This book covers the principles of human cognition, established interaction paradigms, and HCI design and evaluation practices.
- Kahneman, "Thinking, Fast and Slow" [56]: A very accessible summary of Kahneman's seminal research on human decision-making.
- Smith, Goldstein, and Johnson, "Choice Without Awareness: Ethical and Policy Implications of Defaults" [105]: A recent paper discussing the ethical implications of defaults in decision-making. The paper makes suggestions of

how to solve this problem by providing "adaptive defaults" (a type of recommendation).

On research methods

- MacKenzie, "Human-Computer Interaction: An Empirical Research Perspective" [80]: A thorough primer on the design, evaluation and reporting of Human-Computer Interaction experiments.
- Purchase, "Experimental Human-Computer Interaction: A Practical Guide with Visual Examples" [97]: Another primer on experiments; this book contains more details on the evaluation.

On psychometrics and scale development

- DeVellis, "Scale Development, Theory and Applications" [27]: A comprehensive treatment of how to develop measurement scales and assess their quality.
- Schaeffer and Presser, "The Science of Asking Questions" [102]: An in-depth treatment of how to write survey questions.
- Podsakoff, MacKenzie, Lee, and Podsakoff, "Common Method Biases in Behavioral Research" [93]: A paper describing the problem of "Common Method Bias" in survey research, and how to solve or mitigate it.

On statistics

- Utts, "Seeing Through Statistics" [113]: A thorough primer on the statistical evaluation of experimental results.
- Neter, Kutner, Nachtsheim, and Wasserman, "Applied Linear Statistical Models" [86]: A more in-depth treatment of linear statistical methods.
- Kline, "Principles and Practice of Structural Equation Modeling" [59]: An in-depth treatment of structural equation modeling.

We hope that this chapter will spur the adoption of user experiments in the field of recommender systems. We believe that this is an indispensible requirement if the field of recommender systems is indeed to move "from algorithms to user experience" (cf. [72]).

References

1. Adomavicius, G., Kwon, Y.: Improving aggregate recommendation diversity using ranking-based techniques. IEEE Transactions on Knowledge and Data Engineering **24**(5), 896–911 (2012). DOI 10.1109/TKDE.2011.15
2. Ajzen, I.: From intentions to actions: A theory of planned behavior. In: P.D.J. Kuhl, D.J. Beckmann (eds.) Action Control, SSSP Springer Series in Social Psychology, pp. 11–39. Springer Berlin Heidelberg (1985). DOI 10.1007/978-3-642-69746-3_2
3. Ajzen, I.: The theory of planned behavior. Organizational Behavior and Human Decision Processes **50**(2), 179–211 (1991). DOI 10.1016/0749-5978(91)90020-T
4. Ajzen, I., Fishbein, M.: Understanding attitudes and predicting social behaviour. Prentice-Hall, Englewood Cliffs, NJ (1980)

5. Amatriain, X., Pujol, J.M., Tintarev, N., Oliver, N.: Rate it again: Increasing recommendation accuracy by user re-rating. In: Proceedings of the Third ACM Conference on Recommender Systems, RecSys '09, pp. 173–180. ACM, New York, NY, USA (2009). DOI 10.1145/1639714.1639744
6. Basartan, Y.: Amazon versus the shopbot: An experiment about how to improve the shopbots (2001)
7. Bennett, J., Lanning, S.: The netflix prize. In: In KDD Cup and Workshop in conjunction with KDD. San Jose, CA, USA (2007). URL http://www.cs.uic.edu/ liub/KDD-cup-2007/proceedings/The-Netflix-Prize-Bennett.pdf
8. Bentler, P.M., Bonett, D.G.: Significance tests and goodness of fit in the analysis of covariance structures. Psychological Bulletin **88**(3), 588–606 (1980). DOI 10.1037/0033-2909.88.3.588
9. Bettman, J.R., Luce, M.F., Payne, J.W.: Constructive consumer choice processes. Journal of consumer research **25**(3), 187–217 (1998). DOI 10.1086/209535
10. Bilgic, M., Mooney, R.J.: Explaining recommendations: Satisfaction vs. promotion. In: IUI Workshop: Beyond Personalization. San Diego, CA (2005)
11. Blackwelder, W.C.: "Proving the null hypothesis" in clinical trials. Controlled Clinical Trials **3**(4), 345–353 (1982). DOI 10.1016/0197-2456(82)90024-1
12. Bollen, D., Knijnenburg, B.P., Willemsen, M.C., Graus, M.: Understanding choice overload in recommender systems. In: Proceedings of the fourth ACM conference on Recommender systems, pp. 63–70. Barcelona, Spain (2010). DOI 10.1145/1864708.1864724
13. Bollen, K.A.: Structural equation models. In: Encyclopedia of Biostatistics. John Wiley & Sons, Ltd (2005)
14. Bostandjiev, S., O'Donovan, J., Höllerer, T.: TasteWeights: a visual interactive hybrid recommender system. In: Proceedings of the Sixth ACM Conference on Recommender Systems, RecSys '12, pp. 35–42. ACM, Dublin, Ireland (2012). DOI 10.1145/2365952.2365964
15. Cena, F., Vernero, F., Gena, C.: Towards a customization of rating scales in adaptive systems. In: P.D. Bra, A. Kobsa, D. Chin (eds.) User Modeling, Adaptation, and Personalization, no. 6075 in Lecture Notes in Computer Science, pp. 369–374. Springer Berlin Heidelberg (2010). DOI 10.1007/978-3-642-13470-8_34
16. Chen, L., Pu, P.: Interaction design guidelines on critiquing-based recommender systems. User Modeling and User-Adapted Interaction **19**(3), 167–206 (2009). DOI 10.1007/s11257-008-9057-x
17. Chen, L., Pu, P.: Experiments on the preference-based organization interface in recommender systems. ACM Transactions on Computer-Human Interaction **17**(1), 5:1–5:33 (2010). DOI 10.1145/1721831.1721836
18. Chen, L., Pu, P.: Eye-tracking study of user behavior in recommender interfaces. In: P.D. Bra, A. Kobsa, D. Chin (eds.) User Modeling, Adaptation, and Personalization, no. 6075 in Lecture Notes in Computer Science, pp. 375–380. Springer Berlin Heidelberg (2010). DOI 10.1007/978-3-642-13470-8_35
19. Chen, L., Pu, P.: Critiquing-based recommenders: survey and emerging trends. User Modeling and User-Adapted Interaction **22**(1-2), 125–150 (2012). DOI 10.1007/s11257-011-9108-6
20. Chen, L., Tsoi, H.K.: Users' decision behavior in recommender interfaces: Impact of layout design. In: RecSys' 11 Workshop on Human Decision Making in Recommender Systems, pp. 21–26. Chicago, IL, USA (2011). URL http://ceur-ws.org/Vol-811/paper4.pdf
21. Chin, D.N.: Empirical evaluation of user models and user-adapted systems. User Modeling and User-Adapted Interaction **11**(1-2), 181–194 (2001). DOI 10.1023/A:1011127315884
22. Cohen, J.: Statistical power analysis for the behavioral sciences. Psychology Press (1988)
23. Cosley, D., Lam, S.K., Albert, I., Konstan, J.A., Riedl, J.: Is seeing believing?: How recommender system interfaces affect users' opinions. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '03, pp. 585–592. ACM, Ft. Lauderdale, Florida, USA (2003). DOI 10.1145/642611.642713

24. Cramer, H., Evers, V., Ramlal, S., Someren, M., Rutledge, L., Stash, N., Aroyo, L., Wielinga, B.: The effects of transparency on trust in and acceptance of a content-based art recommender. User Modeling and User-Adapted Interaction **18**(5), 455–496 (2008). DOI 10.1007/s11257-008-9051-3
25. Cremonesi, P., Garzotto, F., Negro, S., Papadopoulos, A.V., Turrin, R.: Looking for "Good" recommendations: A comparative evaluation of recommender systems. In: P. Campos, N. Graham, J. Jorge, N. Nunes, P. Palanque, M. Winckler (eds.) Human-Computer Interaction – INTERACT 2011, no. 6948 in Lecture Notes in Computer Science, pp. 152–168. Springer Berlin Heidelberg (2011). DOI 10.1007/978-3-642-23765-2_11
26. Davis, F.D.: Perceived usefulness, perceived ease of use, and user acceptance of information technology. MIS Quarterly **13**(3), 319–340 (1989). DOI 10.2307/249008
27. DeVellis, R.F.: Scale development: theory and applications. SAGE, Thousand Oaks, Calif. (2011)
28. Dooms, S., De Pessemier, T., Martens, L.: An online evaluation of explicit feedback mechanisms for recommender systems. In: 7th International Conference on Web Information Systems and Technologies (WEBIST-2011), pp. 391–394. Noordwijkerhout, The Netherlands (2011). URL https://biblio.ugent.be/publication/2039743/file/2039745.pdf
29. Dooms, S., De Pessemier, T., Martens, L.: A user-centric evaluation of recommender algorithms for an event recommendation system. In: RecSys 2011 Workshop on Human Decision Making in Recommender Systems (Decisions@ RecSys' 11) and User-Centric Evaluation of Recommender Systems and Their Interfaces-2 (UCERSTI 2) affiliated with the 5th ACM Conference on Recommender Systems (RecSys 2011), pp. 67–73. Chicago, IL, USA (2011). URL http://ceur-ws.org/Vol-811/paper10.pdf
30. Downs, J.S., Holbrook, M.B., Sheng, S., Cranor, L.F.: Are your participants gaming the system?: screening mechanical turk workers. In: Proceedings of the 28th SIGCHI conference on Human factors in computing systems, pp. 2399–2402. Atlanta, Georgia, USA (2010). DOI 10.1145/1753326.1753688
31. Ekstrand, M.D., Harper, F.M., Willemsen, M.C., Konstan, J.A.: User perception of differences in recommender algorithms. In: Proceedings of the eighth ACM conference on Recommender systems. Foster City, CA (2014). DOI 10.1145/2645710.2645737
32. Erickson, B.H.: Some problems of inference from chain data. Sociological methodology **10**(1), 276–302 (1979)
33. Farzan, R., Brusilovsky, P.: Encouraging user participation in a course recommender system: An impact on user behavior. Computers in Human Behavior **27**(1), 276–284 (2011). DOI 10.1016/j.chb.2010.08.005
34. Fasolo, B., Hertwig, R., Huber, M., Ludwig, M.: Size, entropy, and density: What is the difference that makes the difference between small and large real-world assortments? Psychology and Marketing **26**(3), 254–279 (2009). DOI 10.1002/mar.20272
35. Faul, F., Erdfelder, E., Lang, A.G., Buchner, A.: G*Power 3: A flexible statistical power analysis program for the social, behavioral, and biomedical sciences. Behavior Research Methods **39**(2), 175–191 (2007). DOI 10.3758/BF03193146
36. Felfernig, A.: Knowledge-based recommender technologies for marketing and sales. Intl. J. of Pattern Recognition and Artificial Intelligence **21**(2), 333–354 (2007). DOI 10.1142/S0218001407005417
37. Fishbein, M., Ajzen, I.: Belief, attitude, intention, and behavior: an introduction to theory and research. Addison-Wesley Pub. Co., Reading, MA (1975)
38. Fisher, R.A.: The design of experiments, vol. xi. Oliver & Boyd, Oxford, England (1935)
39. Freyne, J., Jacovi, M., Guy, I., Geyer, W.: Increasing engagement through early recommender intervention. In: Proceedings of the Third ACM Conference on Recommender Systems, RecSys '09, pp. 85–92. ACM, New York, NY, USA (2009). DOI 10.1145/1639714.1639730
40. Friedrich, G., Zanker, M.: A taxonomy for generating explanations in recommender systems. AI Magazine **32**(3), 90–98 (2011). DOI 10.1609/aimag.v32i3.2365
41. Gedikli, F., Jannach, D., Ge, M.: How should i explain? a comparison of different explanation types for recommender systems. International Journal of Human-Computer Studies **72**(4), 367–382 (2014). DOI 10.1016/j.ijhcs.2013.12.007

42. Gena, C., Brogi, R., Cena, F., Vernero, F.: The impact of rating scales on user's rating behavior. In: D. Hutchison, T. Kanade, J. Kittler, J.M. Kleinberg, F. Mattern, J.C. Mitchell, M. Naor, O. Nierstrasz, C. Pandu Rangan, B. Steffen, M. Sudan, D. Terzopoulos, D. Tygar, M.Y. Vardi, G. Weikum, J.A. Konstan, R. Conejo, J.L. Marzo, N. Oliver (eds.) User Modeling, Adaption and Personalization, vol. 6787, pp. 123–134. Springer, Berlin, Heidelberg (2011). DOI 10.1007/978-3-642-22362-4_11
43. Ghose, A., Ipeirotis, P.G., Li, B.: Designing ranking systems for hotels on travel search engines by mining user-generated and crowdsourced content. Marketing Science **31**(3), 493–520 (2012). DOI 10.1287/mksc.1110.0700
44. Graus, M.P., Willemsen, M.C., Swelsen, K.: The effects of real-time segment-based website adaptation on visitor behavior and visitor experience (2014). Submitted for publication
45. Gregor, S.: The nature of theory in information systems. MIS Quarterly **30**(3), 611–642 (2006). URL http://www.jstor.org/stable/25148742
46. Hassenzahl, M.: The thing and i: understanding the relationship between user and product. In: M. Blythe, K. Overbeeke, A. Monk, P. Wright (eds.) Funology, From Usability to Enjoyment, pp. 31–42. Kluwer Academic Publishers, Dordrecht, The Netherlands (2005). DOI 10.1007/1-4020-2967-5_4
47. Hassenzahl, M.: User experience (UX). In: Proceedings of the 20th International Conference of the Association Francophone d'Interaction Homme-Machine on - IHM '08, pp. 11–15. Metz, France (2008). DOI 10.1145/1512714.1512717
48. Häubl, G., Trifts, V.: Consumer decision making in online shopping environments: The effects of interactive decision aids. Marketing Science **19**(1), 4–21 (2000). URL http://www.jstor.org/stable/193256
49. Heckathorn, D.D.: Respondent-driven sampling II: deriving valid population estimates from chain-referral samples of hidden populations. Social problems **49**(1), 11–34 (2002). DOI 10.1525/sp.2002.49.1.11
50. Herlocker, J.L., Konstan, J.A., Riedl, J.: Explaining collaborative filtering recommendations. In: Proc. of the 2000 ACM conference on Computer supported cooperative work, pp. 241–250. ACM Press, Philadelphia, PA (2000). DOI 10.1145/358916.358995
51. Hu, L., Bentler, P.M.: Cutoff criteria for fit indexes in covariance structure analysis: Conventional criteria versus new alternatives. Structural Equation Modeling: A Multidisciplinary Journal **6**(1), 1–55 (1999). DOI 10.1080/10705519909540118
52. Hu, R., Pu, P.: Enhancing recommendation diversity with organization interfaces. In: Proceedings of the 16th International Conference on Intelligent User Interfaces, IUI '11, pp. 347–350. ACM, Palo Alto, CA, USA (2011). DOI 10.1145/1943403.1943462
53. Iivari, J.: Contributions to the theoretical foundations of systemeering research and the PIOCO model. Ph.D. thesis, University of Oulu, Finland (1983)
54. Jacko, J.A.: The human-computer interaction handbook: fundamentals, evolving technologies, and emerging applications. CRC Press, Boca Raton, FL (2012)
55. Jackson, D.L.: Revisiting sample size and number of parameter estimates: Some support for the n:q hypothesis. Structural Equation Modeling: A Multidisciplinary Journal **10**(1), 128–141 (2003). DOI 10.1207/S15328007SEM1001_6
56. Kahneman, D.: Thinking, fast and slow. Macmillan (2011)
57. Kammerer, Y., Gerjets, P.: How the interface design influences users' spontaneous trustworthiness evaluations of web search results: Comparing a list and a grid interface. In: Proceedings of the 2010 Symposium on Eye-Tracking Research & Applications, ETRA '10, pp. 299–306. ACM, Austin, TX, USA (2010). DOI 10.1145/1743666.1743736
58. Kittur, A., Chi, E.H., Suh, B.: Crowdsourcing user studies with mechanical turk. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 453–456. ACM Press, Florence, Italy (2008). DOI 10.1145/1357054.1357127
59. Kline, R.B.: Principles and practice of structural equation modeling. Guilford Press, New York (2011)
60. Kluver, D., Nguyen, T.T., Ekstrand, M., Sen, S., Riedl, J.: How many bits per rating? In: Proceedings of the Sixth ACM Conference on Recommender Systems, RecSys '12, pp. 99–106. ACM, Dublin, Ireland (2012). DOI 10.1145/2365952.2365974

61. Knijnenburg, B.P.: Simplifying privacy decisions: Towards interactive and adaptive solutions. In: Proceedings of the Recsys 2013 Workshop on Human Decision Making in Recommender Systems (Decisions@ RecSys'13), pp. 40–41. Hong Kong, China (2013). URL http://ceur-ws.org/Vol-1050/paper7.pdf
62. Knijnenburg, B.P., Bostandjiev, S., O'Donovan, J., Kobsa, A.: Inspectability and control in social recommenders. In: Proceedings of the sixth ACM conference on Recommender systems, RecSys '12, pp. 43–50. ACM, Dublin, Ireland (2012). DOI 10.1145/2365952.2365966
63. Knijnenburg, B.P., Kobsa, A.: Making decisions about privacy: Information disclosure in context-aware recommender systems. ACM Transactions on Interactive Intelligent Systems **3**(3), 20:1–20:23 (2013). DOI 10.1145/2499670
64. Knijnenburg, B.P., Kobsa, A., Jin, H.: Dimensionality of information disclosure behavior. International Journal of Human-Computer Studies **71**(12), 1144–1162 (2013). DOI 10.1016/j.ijhcs.2013.06.003
65. Knijnenburg, B.P., Reijmer, N.J., Willemsen, M.C.: Each to his own: how different users call for different interaction methods in recommender systems. In: Proceedings of the fifth ACM conference on Recommender systems, pp. 141–148. ACM Press, Chicago, IL, USA (2011). DOI 10.1145/2043932.2043960
66. Knijnenburg, B.P., Willemsen, M.C.: Understanding the effect of adaptive preference elicitation methods on user satisfaction of a recommender system. In: Proceedings of the third ACM conference on Recommender systems, pp. 381–384. New York, NY (2009). DOI 10.1145/1639714.1639793
67. Knijnenburg, B.P., Willemsen, M.C., Gantner, Z., Soncu, H., Newell, C.: Explaining the user experience of recommender systems. User Modeling and User-Adapted Interaction **22**(4–5), 441–504 (2012). DOI 10.1007/s11257-011-9118-4
68. Knijnenburg, B.P., Willemsen, M.C., Hirtbach, S.: Receiving recommendations and providing feedback: The user-experience of a recommender system. In: F. Buccafurri, G. Semeraro (eds.) E-Commerce and Web Technologies, vol. 61, pp. 207–216. Springer, Berlin, Heidelberg (2010). DOI 10.1007/978-3-642-15208-5_19
69. Knijnenburg, B.P., Willemsen, M.C., Kobsa, A.: A pragmatic procedure to support the user-centric evaluation of recommender systems. In: Proceedings of the fifth ACM conference on Recommender systems, RecSys '11, pp. 321–324. ACM, Chicago, IL, USA (2011). DOI 10.1145/2043932.2043993
70. Kobsa, A., Cho, H., Knijnenburg, B.P.: An attitudinal and behavioral model of personalization at different providers (unpublished manuscript)
71. Köhler, C.F., Breugelmans, E., Dellaert, B.G.C.: Consumer acceptance of recommendations by interactive decision aids: The joint role of temporal distance and concrete versus abstract communications. Journal of Management Information Systems **27**(4), 231–260 (2011). DOI 10.2753/MIS0742-1222270408
72. Konstan, J., Riedl, J.: Recommender systems: from algorithms to user experience. User Modeling and User-Adapted Interaction **22**(1), 101–123 (2012). DOI 10.1007/s11257-011-9112-x
73. Koren, Y., Bell, R., Volinsky, C.: Matrix factorization techniques for recommender systems. Computer **42**(8), 30–37 (2009). DOI 10.1109/MC.2009.263
74. Koren, Y., Sill, J.: OrdRec: An ordinal model for predicting personalized item rating distributions. In: Proceedings of the Fifth ACM Conference on Recommender Systems, RecSys '11, pp. 117–124. ACM, New York, NY, USA (2011). DOI 10.1145/2043932.2043956
75. Landsberger, H.A.: Hawthorne revisited: Management and the worker: its critics, and developments in human relations in industry. Cornell University (1958)
76. Lathia, N., Hailes, S., Capra, L., Amatriain, X.: Temporal diversity in recommender systems. In: Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '10, pp. 210–217. ACM, Geneva, Switzerland (2010). DOI 10.1145/1835449.1835486
77. Lee, Y.E., Benbasat, I.: The influence of trade-off difficulty caused by preference elicitation methods on user acceptance of recommendation agents across loss and gain conditions. Information Systems Research **22**(4), 867–884 (2011). DOI 10.1287/isre.1100.0334

78. Lopes, C.S., Rodrigues, L.C., Sichieri, R.: The lack of selection bias in a snowball sampled case-control study on drug abuse. International journal of epidemiology **25**(6), 1267–1270 (1996). DOI 10.1093/ije/25.6.1267
79. MacCallum, R.C., Widaman, K.F., Zhang, S., Hong, S.: Sample size in factor analysis. Psychological Methods **4**(1), 84–99 (1999). DOI 10.1037/1082-989X.4.1.84
80. MacKenzie, I.S.: Human-Computer Interaction: An Empirical Research Perspective, 1st edn. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (2013)
81. Martin, F.J.: Recsys'09 industrial keynote: Top 10 lessons learned developing deploying and operating real-world recommender systems. In: Proceedings of the Third ACM Conference on Recommender Systems, RecSys '09, pp. 1–2. ACM, New York, NY, USA (2009). DOI 10.1145/1639714.1639715
82. McNee, S.M., Albert, I., Cosley, D., Gopalkrishnan, P., Lam, S.K., Rashid, A.M., Konstan, J.A., Riedl, J.: On the recommending of citations for research papers. In: Proceedings of the 2002 ACM conference on Computer supported cooperative work, pp. 116–125. New Orleans, LA (2002). DOI 10.1145/587078.587096
83. McNee, S.M., Riedl, J., Konstan, J.A.: Being accurate is not enough: how accuracy metrics have hurt recommender systems. In: Extended abstracts on Human factors in computing systems, pp. 1097–1101. Montréal, Québec, Canada (2006). DOI 10.1145/1125451.1125659
84. McNee, S.M., Riedl, J., Konstan, J.A.: Making recommendations better: An analytic model for human-recommender interaction. In: Extended Abstracts on Human Factors in Computing Systems, CHI EA '06, pp. 1103–1108. ACM, Montréal, Québec, Canada (2006). DOI 10.1145/1125451.1125660
85. Mogilner, C., Rudnick, T., Iyengar, S.S.: The mere categorization effect: How the presence of categories increases choosers' perceptions of assortment variety and outcome satisfaction. Journal of Consumer Research **35**(2), 202–215 (2008). DOI 10.1086/586908
86. Neter, J., Kutner, M.H., Nachtsheim, C.J., Wasserman, W.: Applied linear statistical models, vol. 4. Irwin Chicago (1996)
87. Nguyen, T.T., Kluver, D., Wang, T.Y., Hui, P.M., Ekstrand, M.D., Willemsen, M.C., Riedl, J.: Rating support interfaces to improve user experience and recommender accuracy. In: Proceedings of the 7th ACM Conference on Recommender Systems, RecSys '13, pp. 149–156. ACM, Hong Kong, China (2013). DOI 10.1145/2507157.2507188
88. Nuzzo, R.: Scientific method: Statistical errors. Nature **506**(7487), 150–152 (2014). DOI 10.1038/506150a
89. Oestreicher-Singer, G., Sundararajan, A.: Recommendation networks and the long tail of electronic commerce. Management Information Systems Quarterly **36**(1), 65–83 (2012). URL http://aisel.aisnet.org/misq/vol36/iss1/7
90. Oestreicher-Singer, G., Sundararajan, A.: The visible hand? demand effects of recommendation networks in electronic markets. Management Science **58**(11), 1963–1981 (2012). DOI 10.1287/mnsc.1120.1536
91. Orne, M.T.: On the social psychology of the psychological experiment: With particular reference to demand characteristics and their implications. American Psychologist **17**(11), 776–783 (1962). DOI 10.1037/h0043424
92. Paolacci, G., Chandler, J., Ipeirotis, P.: Running experiments on amazon mechanical turk. Judgment and Decision Making **5**(5), 411–419 (2010). URL http://www.sjdm.org/journal/10/10630a/jdm10630a.pdf
93. Podsakoff, P.M., MacKenzie, S.B., Lee, J.Y., Podsakoff, N.P.: Common method biases in behavioral research: A critical review of the literature and recommended remedies. Journal of Applied Psychology **88**(5), 879–903 (2003). DOI 10.1037/0021-9010.88.5.879
94. Pu, P., Chen, L.: Trust-inspiring explanation interfaces for recommender systems. Knowledge-Based Systems **20**(6), 542–556 (2007). DOI 10.1016/j.knosys.2007.04.004
95. Pu, P., Chen, L., Hu, R.: A user-centric evaluation framework for recommender systems. In: Proceedings of the Fifth ACM Conference on Recommender Systems, RecSys '11, pp. 157–164. ACM, Chicago, IL, USA (2011). DOI 10.1145/2043932.2043962

96. Pu, P., Chen, L., Hu, R.: Evaluating recommender systems from the user's perspective: survey of the state of the art. User Modeling and User-Adapted Interaction **22**(4), 317–355 (2012). DOI 10.1007/s11257-011-9115-7
97. Purchase, H.C.: Experimental Human-Computer Interaction: A Practical Guide with Visual Examples, 1st edn. Cambridge University Press, New York, NY, USA (2012)
98. Randall, T., Terwiesch, C., Ulrich, K.T.: User design of customized products. Marketing Science **26**(2), 268–280 (2007). DOI 10.1287/mksc.1050.0116
99. Said, A., Fields, B., Jain, B.J., Albayrak, S.: User-centric evaluation of a k-furthest neighbor collaborative filtering recommender algorithm. In: Proceedings of the 2013 Conference on Computer Supported Cooperative Work, CSCW '13, pp. 1399–1408. ACM, New York, NY, USA (2013). DOI 10.1145/2441776.2441933
100. Said, A., Jain, B.J., Narr, S., Plumbaum, T., Albayrak, S., Scheel, C.: Estimating the magic barrier of recommender systems: A user study. In: Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '12, pp. 1061–1062. ACM, Portland, Oregon (2012). DOI 10.1145/2348283.2348469
101. Salganik, M.J., Heckathorn, D.D.: Sampling and estimation in hidden populations using respondent-driven sampling. Sociological Methodology **34**(1), 193–240 (2004). DOI 10.1111/j.0081-1750.2004.00152.x
102. Schaeffer, N.C., Presser, S.: The science of asking questions. Annual Review of Sociology **29**(1), 65–88 (2003). DOI 10.1146/annurev.soc.29.110702.110112
103. Scheibehenne, B., Greifeneder, R., Todd, P.M.: Can there ever be too many options? a Meta-Analytic review of choice overload. Journal of Consumer Research **37**(3), 409–425 (2010). DOI 10.1086/651235
104. Sinha, R., Swearingen, K.: Comparing recommendations made by online systems and friends. In: In Proceedings of the DELOS-NSF Workshop on Personalization and Recommender Systems in Digital Libraries (2001)
105. Smith, N.C., Goldstein, D.G., Johnson, E.J.: Choice without awareness: Ethical and policy implications of defaults. Journal of Public Policy & Marketing **32**(2), 159–172 (2013). DOI 10.1509/jppm.10.114
106. Sparling, E.I., Sen, S.: Rating: How difficult is it? In: Proceedings of the Fifth ACM Conference on Recommender Systems, RecSys '11, pp. 149–156. ACM, Chicago, IL, USA (2011). DOI 10.1145/2043932.2043961
107. Steele-Johnson, D., Beauregard, R.S., Hoover, P.B., Schmidt, A.M.: Goal orientation and task demand effects on motivation, affect, and performance. Journal of Applied Psychology **85**(5), 724–738 (2000). DOI 10.1037/0021-9010.85.5.724
108. Symeonidis, P., Nanopoulos, A., Manolopoulos, Y.: Providing justifications in recommender systems. IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans **38**(6), 1262–1272 (2008). DOI 10.1109/TSMCA.2008.2003969
109. Tam, K.Y., Ho, S.Y.: Web personalization: is it effective? IT Professional **5**(5), 53–57 (2003). DOI 10.1109/MITP.2003.1235611
110. Tintarev, N., Masthoff, J.: A survey of explanations in recommender systems. In: Data Engineering Workshop, pp. 801–810. IEEE, Istanbul, Turkey (2007). DOI 10.1109/ICDEW.2007.4401070
111. Tintarev, N., Masthoff, J.: Evaluating the effectiveness of explanations for recommender systems. User Modeling and User-Adapted Interaction **22**(4–5), 399–439 (2012). DOI 10.1007/s11257-011-9117-5
112. Torres, R., McNee, S.M., Abel, M., Konstan, J.A., Riedl, J.: Enhancing digital libraries with TechLens+. In: Proceedings of the 2004 joint ACM/IEEE conference on Digital libraries - JCDL '04, pp. 228–236. Tuscon, AZ, USA (2004). DOI 10.1145/996350.996402
113. Utts, J.: Seeing Through Statistics. Cengage Learning (2004)
114. Van Velsen, L., Van Der Geest, T., Klaassen, R., Steehouder, M.: User-centered evaluation of adaptive and adaptable systems: a literature review. The Knowledge Engineering Review **23**(03), 261–281 (2008). DOI 10.1017/S0269888908001379

115. Vargas, S., Castells, P.: Rank and relevance in novelty and diversity metrics for recommender systems. In: Proceedings of the Fifth ACM Conference on Recommender Systems, RecSys '11, pp. 109–116. ACM, Chicago, IL, USA (2011). DOI 10.1145/2043932.2043955
116. Venkatesh, V., Morris, M.G., Davis, G.B., Davis, F.D.: User acceptance of information technology: Toward a unified view. MIS Quarterly **27**(3), 425–478 (2003). URL http://www.jstor.org/stable/30036540
117. Vig, J., Sen, S., Riedl, J.: Tagsplanations: Explaining recommendations using tags. In: Proceedings of the 14th International Conference on Intelligent User Interfaces, IUI '09, pp. 47–56. ACM, Sanibel Island, Florida, USA (2009). DOI 10.1145/1502650.1502661
118. Wang, H.C., Doong, H.S.: Argument form and spokesperson type: The recommendation strategy of virtual salespersons. International Journal of Information Management **30**(6), 493–501 (2010). DOI 10.1016/j.ijinfomgt.2010.03.006
119. Wang, W., Benbasat, I.: Recommendation agents for electronic commerce: Effects of explanation facilities on trusting beliefs. Journal of Management Information Systems **23**(4), 217–246 (2007). DOI 10.2753/MIS0742-1222230410
120. Willemsen, M.C., Graus, M.P., Knijnenburg, B.P.: Understanding the role of latent feature diversification on choice difficulty and satisfaction (manuscript, under review)
121. Willemsen, M.C., Knijnenburg, B.P., Graus, M.P., Velter-Bremmers, L.C., Fu, K.: Using latent features diversification to reduce choice difficulty in recommendation lists. In: RecSys'11 Workshop on Human Decision Making in Recommender Systems, CEUR-WS, vol. 811, pp. 14–20. Chicago, IL (2011). URL http://ceur-ws.org/Vol-811/paper3.pdf
122. Xiao, B., Benbasat, I.: E-commerce product recommendation agents: Use, characteristics, and impact. Mis Quarterly **31**(1), 137–209 (2007). URL http://www.jstor.org/stable/25148784
123. Xiao, B., Benbasat, I.: Research on the use, characteristics, and impact of e-commerce product recommendation agents: A review and update for 2007–2012. In: F.J. Martínez-López (ed.) Handbook of Strategic e-Business Management, Progress in IS, pp. 403–431. Springer Berlin Heidelberg (2014). DOI 10.1007/978-3-642-39747-9_18
124. Zhang, M., Hurley, N.: Avoiding monotony: Improving the diversity of recommendation lists. In: Proceedings of the 2008 ACM Conference on Recommender Systems, RecSys '08, pp. 123–130. ACM, Lausanne, Switzerland (2008). DOI 10.1145/1454008.1454030
125. Zhou, T., Kuscsik, Z., Liu, J.G., Medo, M., Wakeling, J.R., Zhang, Y.C.: Solving the apparent diversity-accuracy dilemma of recommender systems. Proceedings of the National Academy of Sciences **107**(10), 4511–4515 (2010). DOI 10.1073/pnas.1000488107
126. Ziegler, C.N., McNee, S.M., Konstan, J.A., Lausen, G.: Improving recommendation lists through topic diversification. In: Proceedings of the 14th international conference on World Wide Web - WWW '05, pp. 22–32. Chiba, Japan (2005). DOI 10.1145/1060745.1060754