Operations Research

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

Learning Preferences Under Noise and Loss Aversion: An Optimization Approach

Dimitris Bertsimas, Allison O'Hair

OPERATIONS RESEARCH
Vol. 61, No. 5, September–October 2013, pp. 1190–1199
ISSN 0030-364X (print) | ISSN 1526-5463 (online)

http://dx.doi.org/10.1287/opre.2013.1209

Learning Preferences Under Noise and Loss Aversion: An Optimization Approach

Dimitris Bertsimas, Allison O'Hair
Sloan School of Management and Operations Research Center, Massachusetts Institute of Technology, Cambridge, Massachusetts 02142
{dbertsim@mit.edu, akohair@mit.edu}

Preference learning has been a topic of research in many fields, including operations research, marketing, machine learning, and behavioral economics. In this work, we strive to combine the ideas from these different fields into a single methodology to learn preferences and make decisions. We use robust and integer optimization in an adaptive and dynamic way to determine preferences from data that are consistent with human behavior. We use integer optimization to address human inconsistency, robust optimization and conditional value at risk (CVaR) to address loss aversion, and adaptive conjoint analysis and linear optimization to frame the questions to learn preferences. The paper makes the following methodological contributions: to the robust optimization literature by proposing a method to derive uncertainty sets from adaptive questionnaires, to the marketing literature by using the analytic center of discrete sets (as opposed to polyhedra) to capture errors and inconsistencies, and to the risk modeling literature by using efficient methods from computer science for sampling to optimize CVaR. We have implemented an online software that uses the proposed approach and report empirical evidence of its strength.

Subject classifications: programming; integer; linear; applications.
Area of review: Optimization
History: Received November 2012; revision received March 2013; accepted June 2013.

1. Introduction

Learning preferences or utilities for items has been a topic of interest for many years. The goal is to learn user preferences or utilities for a set of items given limited information, with the final outcome being a recommendation made to the user for the item with the highest utility.

Some of the earliest work in preference learning was in the field of economics, in which the emphasis was placed on learning how people behave when faced with a choice between different options. One of the original theories was that of expected utility theory, which claimed that rational people maximize the expectation of a utility function (Bernoulli 1738, von Neumann and Morgenstern 1944). Since then, there have been several influential works disputing the claims of expected utility theory, and instead claiming that users behave in what could be considered "irrational" ways (Kahneman and Tversky 1979, Tversky and Kahneman 1991, Allais 1953). Evidence given in these works shows that when describing their preferences, users are often inconsistent, loss averse, influenced by the framing of the questions, and define outcomes with respect to a reference point.

The field of preference learning has also recently become more popular in machine learning and operations research because of its importance in many Web applications, including search and recommendation systems (Furnkranz and Hullermeier 2011, Doyle 2004). Contrary to the economics literature, the emphasis here is not placed on observing how people behave, but on how to build a preference model given limited data on preferences. This model is often built to predict revenues from offering a subset of products to customers, or to rank a fixed set of options (Farias et al. 2013, Hullermeier et al. 2008). Preference learning is also used to find weights for multiple criteria optimization problems, or multiple criteria decision-making (MCDM) problems (Zionts and Wallenius 1976). Several interactive approaches to learning weights in MCDM problems have been proposed (Miettinen et al. 2008), and there is some work on incorporating robustness into these approaches (Roy 1998, Deb and Gupta 2005, Wang and Zionts 2006). However, there has been limited work on incorporating human behavior into the preference learning and final weight selection (Dyer et al. 1992, Wallenius et al. 2008).

Preference learning is also very popular in marketing, in which preferences are typically learned through questionnaires (Toubia et al. 2003; Carroll and Green 1995; Green and Srinivasan 1978, 1990). The understanding of consumer preferences is a central problem in marketing, and the most widely used method for doing so is by using conjoint analysis or choice questionnaires.

Although there have been many methods proposed to learn preferences, we feel that there is a need for a systematic and comprehensive methodology to algorithmically derive preferences and ultimately make suggestions to users that adhere to human behavior. In previous work, only one utility function or preference order is typically assumed as a result of the preference learning process, but there are often several that are consistent with the known data. We propose a methodology to robustly collect preference data from individuals, as well as ultimately make decisions using robust optimization. Since the selection of a single utility function or ranking order is often arbitrary, robust optimization improves on this approach by using all of the known information to make a decision. Our approach is based on robust and integer optimization, conjoint analysis, and risk measures.

Our overall approach is as follows. We assume that any item or possible outcome $\mathbf{x}$ can be defined by attributes $x_1, \ldots, x_n$. We make the common assumption of a linear utility function $u(\mathbf{x}) = \mathbf{u}'\mathbf{x}$, for which we are trying to learn the weights $\mathbf{u}$. Prior to asking any questions, the weights $\mathbf{u}$ belong in an initial uncertainty set $\mathcal{U}^0 = [-1, 1]^n$, that is we consider a family of possible utilities

$$u(\mathbf{x}) = \mathbf{u}'\mathbf{x}, \quad \mathbf{u} \in \mathcal{U}^0.$$

We present the user with two items in the first question, $\mathbf{x}^1$ and $\mathbf{y}^1$, and ask the user to compare them, i.e., to tell us if he prefers item $\mathbf{x}^1$, prefers item $\mathbf{y}^1$, or has no preference (is indifferent). Based on his answer, we update the uncertainty set and use mixed-integer linear optimization to adaptively generate two new items to ask about in question two, $\mathbf{x}^2$ and $\mathbf{y}^2$. After a number k of such adaptively chosen questions, we have "decreased" the uncertainty set from $\mathcal{U}^0$ to $\mathcal{U}^k \subseteq \mathcal{U}^0$. At this point, we have a family of utilities

$$u(\mathbf{x}) = \mathbf{u}'\mathbf{x}, \quad \mathbf{u} \in \mathcal{U}^k.$$

The idea of using an adaptive questionnaire comes from the marketing literature (Toubia et al. 2003, 2004). Adaptive questionnaires are increasingly being used, but the results often suffer from response errors to early questions that influence the selection of later questions. Some work has been done using complexity control or a Bayesian framework to be robust to response error (Abernethy et al. 2008, Toubia et al. 2007). We instead approach the problem using integer and robust optimization.

Note that we address two separate types of potential noise in the data: human inconsistency and response error. Inconsistency refers to contradictory responses even though all of the responses are accurate. Response error refers to incorrect responses, which may not cause inconsistencies, but will increase the difficulty of capturing the user's true utilities. For these reasons, we also ask the user to indicate if they "feel strongly" about their response to a question to more accurately capture the incorrect answers.

To model loss aversion, we propose to solve the following robust optimization problem:

$$\max_{\mathbf{x}\in\mathbf{X}} \min_{\mathbf{u}\in\mathcal{U}^k} \mathbf{u}'\mathbf{x},$$

where X is the feasible space of outcomes, and k is the number of questions that have been asked. Note that we are taking the perspective that the adaptive process has reduced uncertainty to the set $\mathcal{U}^k$, and, as we are loss averse, we are optimizing the worst utility within the uncertainty set $\mathcal{U}^k$.

Additionally, we propose a robust optimization method to control the trade-off between robustness and optimality. In this method, we maximize the conditional value at risk ($CVaR_\alpha$) of the utility function, which is defined to be the expected value of the worst $\alpha\%$ of the utilities (Rockafellar and Uryasev 2000, Bertsimas et al. 2004, Krzemienowski 2009, Ogryczak 2014). Since the feasible set $\mathcal{U}^k$ is a projection of a mixed-integer set and therefore $CVaR_\alpha$ of the set is defined by an integral, we use random sampling of $\mathcal{U}^k$ to approximate $CVaR_\alpha$. Specifically, we use the "hit-and-run" method of randomly sampling points from a convex body in $\mathbb{R}^n$, introduced by Smith (1984). This method has been shown to perform well in practice, and allows us to efficiently optimize $CVaR_\alpha$ (Vempala 2005).

Artzner et al. (1999) formalized the idea of risk by defining *coherent risk measures*, and $CVaR_\alpha$ is known as the fundamental coherent risk measure. $CVaR_\alpha$ is known as a second-order quantile risk measure, a concept that has been introduced in many ways by many different authors (Artzner et al. 1999, Embrechts et al. 1997, Ogryczak 1998, Rockafellar and Uryasev 2000). We use $CVaR_\alpha$ and not standard deviation or VaR_α since we want to capture the amount of losses incurred. Standard deviation measures variation in both losses and gains, and value at risk (VaR_α), or the α-quantile of the utilities, only captures the number of times you lose, not the severity of the losses. Additionally, neither standard deviation or VaR_α are coherent risk measures, meaning that they violate basic properties that capture the idea of risk. By adjusting the value of α, we are able to select how conservative we would like our robust solution to be.

Overall, we model human behavior in three main ways. First, we model inconsistent behavior and account for the importance of question framing in the way in which we learn individual preferences. Second, we incorporate loss aversion when making decisions for the individual. Lastly, we account for reference dependence in the overall design of the system. A quick and easy questionnaire to learn preferences means that we can ask the questionnaire repeatedly over time as the user's reference point changes.

In summary, our method uses integer optimization, robust optimization and $CVaR_\alpha$, and adaptive conjoint analysis and linear optimization. In our empirical study at the end of this paper, we use $CVaR_\alpha$ to compute the loss aversion of each method quantitatively. We feel this paper makes the following methodological contributions: (a) to the robust optimization literature by proposing to derive uncertainty sets from adaptive questionnaires, (b) to the marketing literature by using the analytic center of discrete sets (as opposed to polyhedra) to capture inconsistencies and errors, and (c) to the risk modeling literature by using efficient methods from computer science for sampling to optimize $CVaR_\alpha$.

This paper is structured as follows. In §2, we introduce the adaptive questionnaire to inform the feasible set of utilities and the mixed-integer optimization model to address human inconsistencies and response error. In §3, we present the robust optimization approach to address loss aversion and the new *CVaR* approach to provide less conservative robust solutions. In §4, we give some empirical evidence to support the use of our strategy in practice. Finally, in §5, we provide some concluding remarks.

2. Building Self-Correcting Utilities Using Adaptive Questionnaires

In this section, we will address how we adaptively ask the user questions to learn a small set of possible utilities, while using integer optimization to account for the inconsistent behavior and response errors of people.

We will denote items by vectors of attributes with superscripts indicating the question. Thus, $\mathbf{x}^i = (x_1^i, x_2^i, \dots, x_n^i)$ is the vector of attributes for one item (item $\mathbf{x}$) that is asked about in question i. The goal of the optimization problem is to suggest the best item for the user, given the utilities $\mathbf{u} = (u_1, u_2, \dots, u_n)$ for the attributes of the items. For example, the items could be different recipes, for which the attributes are the ingredients in the recipes, or the items could be different cars, for which the attributes are different features that can be selected (leather seats, navigation system, etc.). We will assume that there is a finite set of possible attributes, and that the user's utility function is linear in the attributes, $u(\mathbf{x}) = \mathbf{u}'\mathbf{x}$. Suppose we ask the user about two items in question i, $\mathbf{x}^i$ and $\mathbf{y}^i$. If the user indicates that they prefer $\mathbf{x}^i$ (or $\mathbf{y}^i$), then we say that $\mathbf{x}^i > \mathbf{y}^i$ (or $\mathbf{y}^i > \mathbf{x}^i$). If the user indicates that they are indifferent between the two items, we say that $\mathbf{x}^i = \mathbf{y}^i$. Note that this is the result of the user's response; since the user could make errors in their responses, this is the indicated preference of the user, not necessarily the true preference of the user.

We address the inconsistency and response errors of users through a self-correcting mechanism. This accounts for the fact that although transitivity (if $\mathbf{x} > \mathbf{y}$ and $\mathbf{y} > \mathbf{w}$ then $\mathbf{x} > \mathbf{w}$) is rational and ideal, people are often inconsistent. Additionally, people may incorrectly answer questions, resulting in response errors that add noise to the data. Another possible reason is that the user actually has conflicting preferences. For example, the user might indicate that she likes item $\mathbf{x}$ more than item $\mathbf{y}$ ($\mathbf{x} > \mathbf{y}$), item $\mathbf{y}$ more than item $\mathbf{w}$ ($\mathbf{y} > \mathbf{w}$), but item $\mathbf{w}$ more than item $\mathbf{x}$ ($\mathbf{w} > \mathbf{x}$). In this case, it is impossible to find utilities that are consistent with all three responses.

In the adaptive questionnaire, we start with no information about the user's utilities, and we would like to adaptively learn his or her utilities with a series of comparison questions. Furthermore, we would like to select the comparison questions to ask the user so that the space of different possible utility vectors is reduced as quickly as possible. Since the response to each of the questions is unknown, we would like to ask questions that give us the most information possible regardless of the response. The method that we will describe here builds on the method described in Toubia et al. (2003, 2004), but we use integer optimization to account for inconsistencies and response error in the process of selecting the next question. Additionally, we make some changes to the basic algorithm: we select the next question that minimizes the distance to the analytic center of the remaining feasible space, we add an indifferent option, and instead of selecting a particular utility vector at the end of the algorithm, we keep the entire feasible space when we optimize over the utilities. We will discuss the reasons for these changes at the end of this section.

Suppose that we have already asked the user k comparison questions and received responses of the form $\mathbf{x}^i > \mathbf{y}^i$, $\mathbf{y}^i > \mathbf{x}^i$, or $\mathbf{x}^i = \mathbf{y}^i$. Consider the case where the user indicates that they prefer $\mathbf{x}^i$ to $\mathbf{y}^i$ ($\mathbf{x}^i > \mathbf{y}^i$) in question i. We would like the total utility of item $\mathbf{x}^i$ to be larger than the total utility of item $\mathbf{y}^i$, or $\mathbf{u}'(\mathbf{x}^i - \mathbf{y}^i) > 0$. We model the strict inequality here by using a small number $\epsilon > 0$, so we have the constraint $\mathbf{u}'(\mathbf{x}^i - \mathbf{y}^i) \geqslant \epsilon$. However, we would like to account for the possibility of inconsistencies or response error. We do this by introducing a binary variable ϕ_i for question i. Instead of using the constraint above, we use the constraints

$$\mathbf{u}'(\mathbf{x}^i - \mathbf{y}^i) + (n + \epsilon)\phi_i \geqslant \epsilon,$$
$$\mathbf{u}'(\mathbf{x}^i - \mathbf{y}^i) + (n - \epsilon)\phi_i \leqslant n,$$

for question i if the user indicates that $\mathbf{x}^i > \mathbf{y}^i$, where n is the total number of possible attributes. Similarly, we use the constraints

$$\mathbf{u}'(\mathbf{x}^i - \mathbf{y}^i) - (n + \epsilon)\phi_i \leqslant -\epsilon,$$
$$\mathbf{u}'(\mathbf{x}^i - \mathbf{y}^i) - (n - \epsilon)\phi_i \geqslant -n,$$

for question i if the user indicates that $\mathbf{y}^i > \mathbf{x}^i$. In both sets of constraints, if $\phi_i = 0$, then the utility vector is consistent with the user's response to question i (the first constraint enforces the correct response and the second constraint is redundant). If $\phi_i = 1$, then we "flip" the constraint and assume that the user either introduced an inconsistency or made a response error (the first constraint becomes redundant and the second constraint forces the inequality to flip).

If the user indicates that they are indifferent to question i ($\mathbf{x}^i = \mathbf{y}^i$), then we add the constraints

$$-\epsilon \leqslant \mathbf{u}'(\mathbf{x}^i - \mathbf{y}^i) \leqslant \epsilon.$$

In this case, we do not allow the inequalities to flip, since we assume that an indifferent response is more likely to be truthful. Even if the user actually prefers one item over the other, the difference in utilities is probably small, which is captured here. If an indifferent response is given for question i, we set $\phi_i = 0$. We will set $\epsilon = 0.1$ for the empirical evidence given in §4, but our model is robust to the value of ϵ.

Since we expect the user to respond incorrectly to only a small fraction of the questions, we also use the constraint

$$\sum_{i=1}^{k} \phi_i \leqslant \gamma k,$$

where γ is a parameter that indicates the maximum fraction of responses that we allow to be incorrect. For example, if $\gamma = 0.1$, we allow at most 10% of the user's responses to be flipped. In this work, we only give an upper bound on the sum of the ϕ_i variables. This is due to the desire to be robust to a certain number of response errors and inconsistencies that could occur. An alternative would be to add a term to the objective that penalizes "flipping." However, this would minimize the number of response errors we assume happen, instead of the number of response errors that actually happen. We would like to be robust to a certain number of response errors potentially occurring in practice.

These constraints provide a description of the feasible space of utilities. Throughout the remainder of this paper, we will also restrict the vector of utilities $\mathbf{u}$ to be in $[-1, 1]^n$, without loss of generality.

We would now like to select the next question to ask using this feasible space of utilities. We do this by finding the analytic center of the feasible set for $\mathbf{u}$, which is the projection of a mixed-integer set. To find the analytic center of this set, we solve the following optimization problem, where we use the notation $\mathbf{z}^i = \mathbf{x}^i - \mathbf{y}^i$ for the difference between the attribute vectors of the items asked about in question i:

$$\text{maximize} \quad \sum_{\substack{i=1\\ x_i \neq y_i}}^{k} \log(s_i) + \sum_{\substack{i=1\\ x_i = y_i}}^{k} [\log(s_i^1) + \log(s_i^2)] + \sum_{j=1}^{2n} \log(t_j) \tag{1}$$

$$\text{s.t.} \quad -\mathbf{u}'\mathbf{z}^i - (n+\epsilon)\phi_i + s_i^1 = -\epsilon, \quad \forall i \in \{1, \ldots, k\} \text{ s.t. } \mathbf{x}^i > \mathbf{y}^i, \tag{1a}$$

$$\mathbf{u}'\mathbf{z}^i + (n-\epsilon)\phi_i + s_i^2 = n, \quad \forall i \in \{1, \ldots, k\} \text{ s.t. } \mathbf{x}^i > \mathbf{y}^i, \tag{1b}$$

$$\mathbf{u}'\mathbf{z}^i - (n+\epsilon)\phi_i + s_i^1 = -\epsilon, \quad \forall i \in \{1, \ldots, k\} \text{ s.t. } \mathbf{y}^i > \mathbf{x}^i, \tag{1c}$$

$$-\mathbf{u}'\mathbf{z}^i + (n-\epsilon)\phi_i + s_i^2 = n, \quad \forall i \in \{1, \ldots, k\} \text{ s.t. } \mathbf{y}^i > \mathbf{x}^i, \tag{1d}$$

$$\mathbf{u}'\mathbf{z}^i + s_i^1 = \epsilon, \quad \forall i \in \{1, \ldots, k\} \text{ s.t. } \mathbf{x}^i = \mathbf{y}^i, \tag{1e}$$

$$-\mathbf{u}'\mathbf{z}^i + s_i^2 = \epsilon, \quad \forall i \in \{1, \ldots, k\} \text{ s.t. } \mathbf{x}^i = \mathbf{y}^i, \tag{1f}$$

$$\phi_i = 0, \quad \forall i \in \{1, \ldots, k\} \text{ s.t. } \mathbf{x}^i = \mathbf{y}^i, \tag{1g}$$

$$-M\phi_i \leqslant s_i - s_i^1 \leqslant M\phi_i, \quad i = 1, \ldots, k \text{ s.t. } \mathbf{x}^i \neq \mathbf{y}^i, \tag{1h}$$

$$-M(1-\phi_i) \leqslant s_i - s_i^2 \leqslant M(1-\phi_i), \quad i = 1, \ldots, k \text{ s.t. } \mathbf{x}^i \neq \mathbf{y}^i, \tag{1i}$$

$$-u_j + t_j = 1, \quad j = 1, \ldots, n, \tag{1j}$$

$$u_j + t_{n+j} = 1, \quad j = 1, \ldots, n, \tag{1k}$$

$$\sum_{i=1}^{k} \phi_i \leqslant \gamma k, \tag{1l}$$

$$\phi_i \in \{0, 1\}, \quad i = 1, \ldots, k, \tag{1m}$$

$$s_i \geqslant 0, \quad i = 1, \ldots k, \tag{1n}$$

$$t_j \geqslant 0, \quad j = 1, \ldots 2n. \tag{1o}$$

The objective of (1) is to maximize the log of the slack variables, as defined by the analytic center (Boyd and Vandenberghe 2004). Constraints (1a)–(1g) represent the question constraints defined previously in this section, but with slack variables added in. Note that the slack variables are slightly more complicated than those in a typical optimization problem for the questions for which the user makes a choice. In our formulation, each question defines two constraints, but only one of them is nontrivial, depending on the value of ϕ_i. If $\phi_i = 0$, then the first question constraint, (1a) or (1c), defines the feasible utilities for that question, so we would only like to consider the slack variable for the first constraint (s_i^1), when maximizing the sum of the log of the slacks. If $\phi_i = 1$, then the second question constraint, (1b) or (1d), defines the feasible utilities for that question, so we would only like to consider the slack variable for the second constraint (s_i^2), when maximizing the sum of the log of the slacks. Constraints (1h)–(1i) use the "big-M" approach to set $s_i = s_i^1$ or $s_i = s_i^2$, depending on the value of ϕ_i, where s_i is the slack variable that contributes to the objective. The remaining constraints define the feasible region as described previously.

In practice, we solve this problem by first finding a strictly interior feasible point and then using Newton's method (or a suitable approximation of Newton's method) to iteratively compute the analytic center (for a more detailed explanation, Toubia et al. 2003, 2004). Note, however, that unlike Toubia et al. (2003, 2004), we compute the analytic center of a set that involves continuous and discrete variables.

Denote the optimal solution for $\mathbf{u}$, or the analytic center, by $\mathbf{c}^*$. To try and cut the feasible region as much as possible, we select as the next question the one whose hyperplane is as close to $\mathbf{c}^*$ as possible. This is the solution to the following problem:

$$i^* = \arg\min_i \frac{|(\mathbf{c}^*)'\mathbf{z}^i|}{\|\mathbf{z}^i\|},$$

where the minimization is over all possible questions $\mathbf{z}^i = \mathbf{x}^i - \mathbf{y}^i$ between two items $\mathbf{x}^i$ and $\mathbf{y}^i$ that can be asked. Then question $k+1$ is defined by $\mathbf{z}^{i^*}$. Note that although this problem requires an enumeration of all possible question pairs ($O(n^2)$), it is fast for reasonably sized practical problems (hundreds of items and attributes). After asking question $k+1$ and receiving the response, we add two new question constraints and a variable ϕ_i to (1) and repeat the procedure described here to find question $k+2$.

At any point, we may want to stop asking questions, or the remaining feasible region might be so small that it is impractical or unproductive to continue asking questions. A common strategy is to compute the analytic center one last time, and use this as an estimate for the utilities. We propose a new approach that uses robust optimization over the entire feasible set, which we describe in §3.

In addition to the methodological change of finding the analytic center of a mixed continuous and discrete set, we made a few additional changes to the methodology proposed in Toubia et al. (2003, 2004). The first is that we select the next question that minimizes the distance to the analytic center of the remaining feasible space, instead of selecting a hyperplane that goes through the analytic center and is parallel to the shortest axis of the bounding ellipse as proposed by Toubia et al. (2003, 2004). The reason for this change is that we assume that we have a fixed set of items that we can ask about. We do not have the freedom to construct the best possible item to ask about, but instead have to pick from one of the fixed options. This means that we are not able to ask a question that necessarily goes through the analytic center of the polyhedron and is parallel to the shortest axis of a bounding ellipse. By selecting the question that has a hyperplane closest to the analytic center, we are using a variation of the idea proposed by Toubia et al. (2003, 2004), with the same goal of reducing the feasible space by as much as possible with each question, regardless of the response given by the user.

We also add the option for a user to answer that they are "indifferent" between the two items, instead of forcing the user to pick between the two items. This adds an extra level of complexity to the problem, but also makes the questionnaire more user-friendly.

Additionally, instead of selecting a particular utility vector at the end of the algorithm, we keep the entire feasible space when we optimize over the utilities. The reason for this change is that we would like our approach to be more robust to error. We discuss this further in the next section.

2.1. Adding a "Feel Strongly" Option

In (1), we assume that all questions are equally likely to contain response errors (unless the user selected the indifferent option) and thus any question constraints can be flipped. However, it is more likely that the user "feels strongly" about some responses, and is more ambivalent about others. We add an additional option for the user to indicate that they feel strongly about a response, in which case we do not include constraints (1b) or (1d) of (1). By doing this, if $\phi_i = 1$, the only constraint for question i becomes trivial, and thus we are not considering that question when computing the utilities. We do not want to force the inequality to flip, since the user felt strongly about their response, but we may need to relax the constraint because of inconsistencies in the responses.

Thus, when faced with two items, $\mathbf{x}^i$ or $\mathbf{y}^i$, the user can indicate that (a) he prefers $\mathbf{x}^i$ over $\mathbf{y}^i$, (b) he prefers $\mathbf{y}^i$ over $\mathbf{x}^i$, (c) he strongly prefers $\mathbf{x}^i$ over $\mathbf{y}^i$, (d) he strongly prefers $\mathbf{y}^i$ over $\mathbf{x}^i$, or (e) he is indifferent. We will denote a strong preference for item $\mathbf{x}^i$ over item $\mathbf{y}^i$ by $\mathbf{x}^i >> \mathbf{y}^i$.

With this additional option, we solve (1) with constraints (1a)–(1d) replaced by the following constraints:

$$\begin{aligned}
&-\mathbf{u}'\mathbf{z}^i - (n+\epsilon)\phi_i + s_i^1 = -\epsilon, \\
&\qquad\qquad \forall\, i \in \{1,\ldots,k\} \text{ s.t. } \mathbf{x}^i > \mathbf{y}^i \text{ or } \mathbf{x}^i >> \mathbf{y}^i, \\
&\mathbf{u}'\mathbf{z}^i + (n-\epsilon)\phi_i + s_i^2 = n, \\
&\qquad\qquad \forall\, i \in \{1,\ldots,k\} \text{ s.t. } \mathbf{x}^i > \mathbf{y}^i, \\
&\mathbf{u}'\mathbf{z}^i - (n+\epsilon)\phi_i + s_i^1 = -\epsilon, \\
&\qquad\qquad \forall\, i \in \{1,\ldots,k\} \text{ s.t. } \mathbf{y}^i > \mathbf{x}^i \text{ or } \mathbf{y}^i >> \mathbf{x}^i, \\
&-\mathbf{u}'\mathbf{z}^i + (n-\epsilon)\phi_i + s_i^2 = n, \\
&\qquad\qquad \forall\, i \in \{1,\ldots,k\} \text{ s.t. } \mathbf{y}^i > \mathbf{x}^i, \\
&s_i^2 = 0, \quad \forall\, i \in \{1,\ldots,k\} \text{ s.t. } \mathbf{x}^i >> \mathbf{y}^i \text{ or } \mathbf{y}^i >> \mathbf{x}^i.
\end{aligned} \tag{2}$$

Note that we set $s_i^2 = 0$ if the user feels strongly about the response to question i, since we do not have a second constraint. By using the new constraints (2), we are able to correct for inconsistencies or errors in the responses, while decreasing the chance of violating the inequalities that we know the user feels strongly about, and are therefore most likely to be correct.

3. Loss Averse Solutions with Robust Optimization

Up to this point, we have been concerned with estimating utilities. However, our main concern is with selecting the appropriate item in order to maximize a user's utility subject to a set of constraints on the items. This is a common problem in many applications; a decision needs to be made among a set of items, and a natural goal is to maximize utility. This problem can be modeled by the following optimization problem:

$$\begin{aligned}
\text{maximize} \quad & \mathbf{u}'\mathbf{x} \\
\text{s.t.} \quad & \mathbf{x} \in X,
\end{aligned}$$

where X is our feasible set, $\mathbf{u}$ is our utility vector, and each possible choice is modeled by a vector of attributes denoted by $\mathbf{x} \in X$. Note that we have not made any assumptions regarding the feasible set X.

This optimization problem assumes that the utility vector **u** is fixed, but in many situations the utility vector is unknown. In the previous section, we proposed a strategy for learning **u** which, after a number k of questions, gives us a feasible region that is continuous in **u** and discrete in $\boldsymbol{\phi}$. Denote by $\mathcal{U}^k$ the feasible set of **u** for all possible values of $\boldsymbol{\phi}$ after k questions (note that we will not use the feel strongly option throughout this section, but the formulations can easily be extended to include it):

$$\begin{aligned}\mathcal{U}^k = \Big\{\mathbf{u}\in\mathbb{R}^n \;|\; & \mathbf{u}'\mathbf{z}^i + (n+\epsilon)\phi_i \geqslant \epsilon,\\ & \qquad \forall\, i\in\{1,\dots,k\} \text{ s.t. } \mathbf{x}^i > \mathbf{y}^i,\\ & \mathbf{u}'\mathbf{z}^i + (n-\epsilon)\phi_i \leqslant n,\\ & \qquad \forall\, i\in\{1,\dots,k\} \text{ s.t. } \mathbf{x}^i > \mathbf{y}^i,\\ & \mathbf{u}'\mathbf{z}^i - (n+\epsilon)\phi_i \leqslant -\epsilon,\\ & \qquad \forall\, i\in\{1,\dots,k\} \text{ s.t. } \mathbf{y}^i > \mathbf{x}^i,\\ & \mathbf{u}'\mathbf{z}^i - (n-\epsilon)\phi_i \geqslant -n,\\ & \qquad \forall\, i\in\{1,\dots,k\} \text{ s.t. } \mathbf{y}^i > \mathbf{x}^i,\\ & -\epsilon \leqslant \mathbf{u}'\mathbf{z}^i \leqslant \epsilon,\ \forall\, i\in\{1,\dots,k\} \text{ s.t. } \mathbf{x}^i = \mathbf{y}^i,\\ & \phi_i = 0,\ \forall\, i\in\{1,\dots,k\} \text{ s.t. } \mathbf{x}^i = \mathbf{y}^i,\\ & -1 \leqslant u_i \leqslant 1,\ i=1,\dots,n,\\ & \sum_{i=1}^{k}\phi_k \leqslant \gamma k,\\ & \phi_i \in \{0,1\},\ i=1,\dots,k\Big\}.\end{aligned}$$

In previous work, the standard approach was to select a particular estimate for **u** out of all possible choices in $\mathcal{U}^k$, where the most typical choice has been to select the analytic center of $\mathcal{U}^k$. Here, we instead solve a robust optimization problem that considers the entire set $\mathcal{U}^k$:

$$\max_{\mathbf{x}\in X}\left[\min_{\mathbf{u}\in\mathcal{U}^k} \mathbf{u}'\mathbf{x}\right]. \tag{3}$$

For a fixed $\mathbf{x}\in X$, the inner optimization problem selects the worst possible utility vector that is in the set $\mathcal{U}^k$. Since the set $\mathcal{U}^k$ came from the adaptive questionnaire, it is possible that any vector in $\mathcal{U}^k$ is the user's true utility vector, and so we are looking at the worst-case outcome. The outer optimization problem then tries to find the best item $\mathbf{x}\in X$ given this worst-case approach. This problem is thus robust in the sense that we are trying to maximize the worst-case scenario, and so we aim to be robust to error. This approach is typically referred to as Wald's maximin model (Wald 1945) and is one of the most important models in robust optimization. Note that for a given $\mathbf{x}\in X$, the objective of (3) is a concave function, which captures the risk adverse utilities often exhibited by individuals.

Typically, this problem is solved by taking the dual of the inner problem, resulting in an optimization problem that is no more difficult than the original optimization problem (Bertsimas and Sim 2004). However, in our case, the inner problem is a mixed-integer optimization problem, since the ϕ_i variables are binary. Since in many applications, the set X only specifies that one item should be selected, we will solve this problem by enumeration when we report empirical results in §4. Thus, for each possible $\bar{\mathbf{x}}\in X$, we solve

$$\min_{\mathbf{u}\in\mathcal{U}^k} \mathbf{u}'\bar{\mathbf{x}},$$

and then select the **x** for which the objective function value is the largest. This problem can be solved with mixed-integer optimization techniques.

3.1. A Robust Approach Using CVaR

Although the robust optimization approach provides a loss averse method, it tends to produce solutions that may be too conservative. In this section, we present a method to model loss averse behavior with robust optimization using the concept of conditional value at risk (*CVaR*), or expected shortfall (Artzner et al. 1999). *CVaR* is a risk measure often used in finance to evaluate the market risk of a portfolio. The "*CVaR* at the α% level" or $CVaR_\alpha$ is the average value of the worst α% of the cases. It is an alternative to value at risk (VaR_α), the α-quantile, but it is more sensitive to losses (Artzner et al. 1999). Mathematically, *CVaR* can be defined as follows:

$$CVaR_\alpha = \frac{1}{\alpha}\int_0^\alpha VaR_\gamma\, d\gamma.$$

The motivation for this approach is similar to the tradeoff proposed by Bertsimas and Sim (2004), in that we would like to adjust the conservatism of the approach. The analytic center approach often yields better objective function values on average, but also tends to have greater losses in the domain of losses than the robust version (it is not very loss averse). By using *CVaR*, we are able to maintain the best features of both approaches. This will be shown in the empirical evidence in §4.

We would like to maximize the *CVaR* of the possible utility vectors in the space of feasible utilities $\mathcal{U}^k$. Since this region includes discrete values for ϕ_i, we first fix these values to the values at the analytic center, which is found after the final question is asked using the procedure described in §2. We assume that these values capture good values for ϕ_i, but in practice they could be varied between all possible values.

Since precisely modeling *CVaR* would require an integral in the objective, we use random sampling to approximate the problem. Our feasible region $\mathcal{U}^k$ becomes a polytope when we fix the values of ϕ_i. There are several different ways to randomly sample from a polytope (Vempala 2005).

We use the "hit-and-run" algorithm, starting from the analytic center (which has been computed in the computation of $\mathcal{U}^k$) to find N different utility vectors $u_1, \ldots, u_N \in \mathcal{U}^k$. The hit-and-run algorithm was introduced by Smith (1984), and is defined as follows:

- Pick a uniformly distributed random line l through the current point.
- Move to a uniform random point along the chord $l \cap \mathcal{U}^k$.

Smith (1984) proved that the stationary distribution of the hit-and-run walk is the uniform distribution over $\mathcal{U}^k$. Lovász and Vempala (2006) showed that the hit-and-run walk mixes (becomes stationary) in $O(n^4 \ln^3(n/d))$ steps starting from a point at distance d from the boundary, and is thus a polynomial time algorithm. Other random walks, including the well-known "ball walk," can potentially take exponentially many steps from some starting points. In addition to hit-and-run being a polynomial time algorithm, it is also known to perform very well in practice.

Given the random sample of utility vectors $\{u_1, \ldots, u_N\}$, we then maximize the *CVaR* of this representative sample of utility vectors using the following robust optimization problem (Ogryczak 2014):

$$\begin{aligned} \max_{\mathbf{x}\in X} \min_{\mathbf{y}} \quad & \frac{1}{\alpha N}\sum_{j=1}^{N}(\mathbf{u}_j'\mathbf{x})y_j \\ & \sum_{j=1}^{N} y_j = \alpha N \\ & 0 \leqslant y_j \leqslant 1, \quad j=1,\ldots,N, \end{aligned} \tag{4}$$

where α controls how conservative, or loss averse, we would like to be. If we set $\alpha = 1/N$ for $N \to \infty$, this is equivalent to the robust approach.

We can represent this problem as a linear optimization problem by taking the dual of the inner minimization problem (Bertsimas et al. 2011). We define dual variables θ and $\mathbf{w}$. Then, the problem can be reformulated as

$$\begin{aligned} \max_{\mathbf{x},\theta,\mathbf{w}} \quad & \theta + \frac{1}{\alpha N}\sum_{j=1}^{N} w_j \\ & \theta + w_j \leqslant \mathbf{u}_j'\mathbf{x}, \quad j=1,\ldots,N \\ & w_j \leqslant 0, \quad j=1,\ldots,N \\ & x \in X. \end{aligned} \tag{5}$$

Although our approach requires random sampling of the feasible region, it successfully provides loss averse solutions without being overly conservative. We will show the benefits of this formulation in §4.

4. Empirical Evidence

In this section, we present evidence that a self-correcting and robust approach is feasible, realistic, and appealing in practice. We present empirical evidence comparing the robust approach, the analytic center approach, and the CVaR approach.

We have implemented an online software that uses the proposed approach to select preferred recipes for the user of a personalized dieting application. Users are asked to answer comparison questions as described in §2 given the title of the recipes, pictures, descriptions of the recipes, and the ingredients. They are also given the option to indicate that they strongly prefer one recipe over the other. We would like to learn the user's utilities for the ingredients (or other attributes) of the recipes, and then ultimately suggest a meal plan that is appealing to them. In this section, we present empirical evidence supporting our methodology using this online software.

To compare the validity of the different approaches, we performed the following experiment. We first generated a "true" utility vector $\mathbf{u}^*$ that a user could have, by randomly sampling a feasible utility vector from the initial feasible space of utilities $\mathcal{U}^0 = [-1, 1]^n$. Using this true utility vector, we generated the appropriate answers for a fixed number of comparison questions, where the questions were selected using the adaptive questionnaire described in §2. For each question, we inserted some normally distributed noise ζ with mean zero and standard deviation σ, into the response to that question. The reason for this is that we assume people make errors in their responses, either because of inaccurate responses, or ambiguous preferences. Thus, for $\zeta \sim N(0, \sigma)$, if

$$(\mathbf{u}^*)'\mathbf{z}^i + \zeta \geqslant \hat{\epsilon},$$

we record the user's response to question i as $\mathbf{x}^i > \mathbf{y}^i$. If

$$(\mathbf{u}^*)'\mathbf{z}^i + \zeta \leqslant -\hat{\epsilon}$$

we record the user's response to question i as $\mathbf{y}^i > \mathbf{x}^i$. Lastly, if

$$-\hat{\epsilon} \leqslant (\mathbf{u}^*)'\mathbf{z}^i + \zeta \leqslant \hat{\epsilon}$$

we record the user's response to question i as $\mathbf{x}^i = \mathbf{y}^i$.

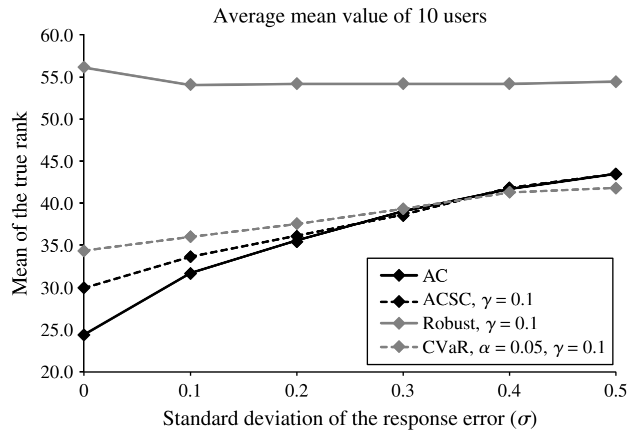
In the empirical evidence presented in this section, we let $\hat{\epsilon} = 0.02$. This causes zero to four indifferent responses for every 10 questions, depending on the true utility vector of the user. We assume here that although some indifferent responses might be made, the user will select between the two options most of the time. We have two reasons for this. The first is that the methodology proposed here is designed to ask comparison questions. If the user is answering indifferent for the majority of the questions, a comparison questionnaire is probably not appropriate for the specific application. The second reason is that our methodology is designed to handle response errors and inconsistencies, partly because we are forcing the user to pick between the two options.

Note that it is the self-correcting property of our approach that insures the feasibility of the algorithm regardless of how much noise is inserted into the responses. Even if inconsistencies are created, with γ large enough, the feasible space $\mathcal{U}^k$ will remain nonempty. However, recall that the adaptive questionnaire is designed so that inconsistencies will be avoided as much as possible, so we expect a small number of inconsistencies, if any. Since the next question is selected as one that cuts the current feasible space as close as possible to the analytic center, it will always leave a remaining feasible space unless the closest cut to the analytic center does not go through the current feasible space at all (leaving an empty feasible space with one response to the question, and the same feasible space as before with the other response to the question). Although unlikely, if this does happen and the new feasible space turns out to be empty, the questionnaire will stop and the most recent nonempty feasible space will be the final feasible space. Throughout the empirical studies performed here, this situation has never occurred.

Using these responses, the ultimate optimization problem (that of finding an appealing meal plan for the user of a personalized dieting system) was then solved using the different methods discussed in this paper: the traditional analytic center approach with "slack" variables to account for inconsistencies, but without the self-correcting mechanism (Toubia et al. 2003, 2004), the new analytic center approach with the self-correcting mechanism presented in (1), the robust approach given by (3), and the *CVaR* approach given by (5). We will denote these methods by AC, ACSC, Robust, and *CVaR*, respectively. Note that there are two different analytic center approaches here—the traditional approach without accounting for inconsistencies and responses errors (AC), and the new approach with the self-correcting mechanism (ACSC).

We report the results of these methods as follows. We first rank all of the items (recipes) according to the "true" utilities, $\mathbf{u}^*$. Thus, the item with the highest true utility gets rank 1, and the item with the lowest utility gets rank 102 (the total number of recipes in the data set). We then find the top five items according to the different optimization methods. The reason for this is that in this application and in many others, we will often suggest several items to the user that we think they will like. In the case of recipes, we will often want to make several suggestions since the user may like many different types of recipes, and we would like to make sure we suggest one that they are interested in on any particular day. Therefore, after solving the optimization problem corresponding to each of the methods, we eliminate the optimal solution, and solve again to get the next best solution. We repeat this three more times to get the top five solutions. We can then compare the selected solutions with the true rank of the items. We report the average true rank of the top five solutions found. Therefore, smaller values are preferred.

Figure 1. Plot of the average values for selective methods and parameters.



Throughout the rest of this section, we report results for 10 different true utility vectors (or 10 different "users"), and 200 runs per user for each value of $\sigma > 0$ to account for the noise in the responses. Our data set consisted of 102 recipes for the main dish at dinner (so that the recipes were comparable), and were described by 186 attributes (the ingredients of the recipes). An additional parameter to consider in this approach is the number of questions to ask. We will report results for 10 questions, since it is a realistic number of questions that might be asked in an application.

The complete numerical results are presented in the appendix (available as supplemental material at http://dx.doi.org/10.1287/opre.2013.1209). Figures 1, 2, and 3 show these results as functions of the response noise σ. The particular parameter choices shown in the figures were selected as representative values that one might use in practice, but can also be selected through cross-validation.

Figure 1 gives the average rank of the top five solutions for AC, ACSC, *CVaR* (with $\alpha = 0.05$), and robust, with $\gamma = 0.1$ for all methods. Recall that there are 102 total recipes, so the average rank can range from 3 to 100, where smaller values are preferred. As can be seen from this figure, the AC, ACSC, and CVaR methods are competitive, with the CVaR method slightly better than the other two for large σ, and the AC method better than the other two for small σ. The ACSC and CVaR methods are also most robust to noise than the AC method. Since these methods better account for response error and inconsistencies, they logically perform better with $\sigma > 0$.

The robust method is strictly dominated by the other methods, but it is very consistent regardless of the amount of noise. Additionally, the numerical results in the appendix give the standard deviation of the methods. The *CVaR* method has a consistently lower standard deviation than the traditional AC method, and the robust method has a significantly lower standard deviation than all other methods. This provides evidence that the new methods are more risk averse than the traditional analytic center approach.

Figure 2. Plot of the CVaR at $\alpha = 5\%$ values for selective methods and parameters.

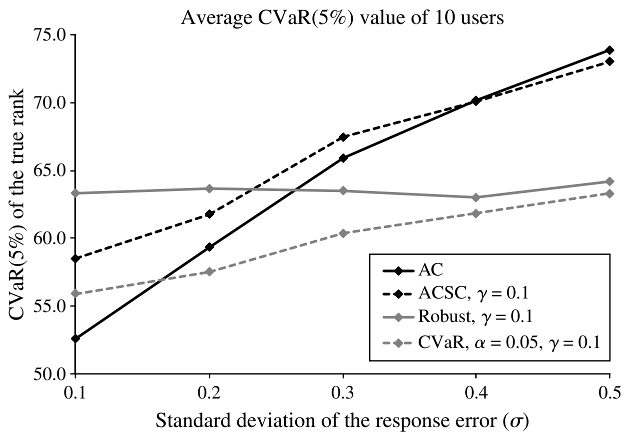
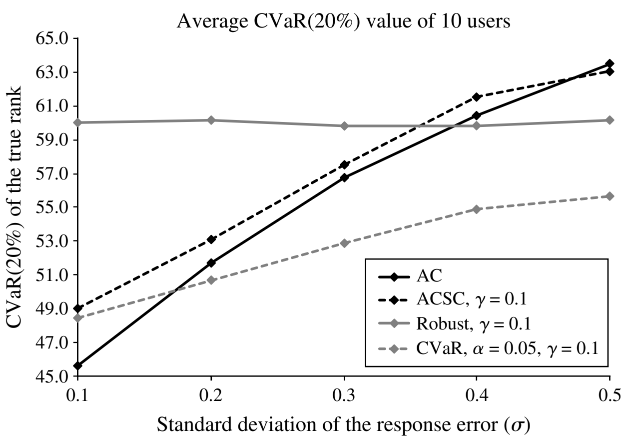


Figure 3. Plot of the CVaR at $\alpha = 20\%$ values for selective methods and parameters.



Figures 2 and 3 give the conditional value at risk (*CVaR*) for $\alpha = 5\%$ and for $\alpha = 20\%$, respectively. We use this as a risk measure to compare how loss averse each of the methods are. The lower the *CVaR* value (meaning that the worst-case solutions have better ranks), the more loss averse a method is considered to be.

Figure 2 shows that for almost all levels of response error, the CVaR method strictly dominates the other methods for the worst 5% of cases. The AC method is slightly better for small response error, and the robust method is competitive for high response error. Figure 3 shows similar behavior, but the robust method is not as strong. This shows us that the CVaR method is able to nicely balance robustness with optimality, as it was designed to do.

When considering all three plots together, we would argue that for small response error ($0 < \sigma \leqslant 0.2$), the AC approach is the best on average, as expected. But the CVaR approach starts looking better as the response error increases in this interval. For moderate to high response error ($0.2 < \sigma \leqslant 0.5$), the CVaR method would be the best choice. It has the best performance when considering both the average value and CVaR. Overall, given that we expect some response error from the users, we would argue that the CVaR approach shows the best numerical results, in terms of losses and average performance.

5. Concluding Remarks

We have developed an optimization-based approach for preference learning that incorporates techniques and observations from many different fields. Our approach addresses some of the key observations of preference learning in behavioral economics: people are loss averse, are inconsistent, and evaluate outcomes with respect to deviations from a reference point. We have shown how mixed binary optimization can be used to correct for inconsistent behavior, choice-based conjoint analysis can be used in an adaptive questionnaire to dynamically select pairwise questions, and robust linear optimization and *CVaR* can model loss averse behavior.

Furthermore, we gave empirical evidence from an online software we developed that strives to model human preferences in a realistic situation. We have shown that the *CVaR* approach in particular performs very well, and is more robust to noise in the responses and is more loss averse than the traditional analytic center approach.

Supplemental Material

Supplemental material to this paper is available at http://dx.doi.org/10.1287/opre.2013.1209.

Acknowledgments

The authors would like to acknowledge Spyros Potamianos and Manolis Tsangaris for their help in implementing the online software.

References

Abernethy J, Evgeniou T, Toubia O, Vert J-P (2008) Eliciting consumer preferences using robust adaptive choice questionnaires. *IEEE Trans. Knowledge & Data Engrg.* 20(2):145–155.

Allais M (1953) Le comportement de l'homme rationnel devant le risque: Critique des postulats et axiomes de l'école américaine. *Econometrica* 21(4):503–546.

Artzner P, Delbaen F, Eber J-M, Heath D (1999) Coherent measures of risk. *Math. Finance* 9(3):203–228.

Bernoulli D (1738) Specimen theoriae novae de mensura sortis. *Comentarii Academiae Scientiarum Imperialis Petropolitanae* 5:175–192. [Translation by Louise Sommer]

Bertsimas D, Sim M (2004) The price of robustness. *Oper. Res.* 52(1):35–53.

Bertsimas D, Brown DB, Caramanis C (2011) Theory and applications of robust optimization. *SIAM Rev.* 53(3):464–501.

Bertsimas D, Lauprete GJ, Samarov A (2004) Shortfall as a risk measure: Properties, optimization and applications. *J. Econom. Dynam. Control* 28(7):1353–1381.

Boyd S, Vandenberghe L (2004) *Convex Optimization* (Cambridge University Press, New York).

Carroll D, Green P (1995) Psychometric methods in marketing research: Part I, conjoint analysis. *J. Marketing Res.* 32:385–391.

Deb K, Gupta H (2005) Searching for robust Pareto-optimal solutions in multi-objective optimization. *Proc. Third Evolutionary Multi-Criterion Optim. (EMO-03).*

Doyle J (2004) Prospects for preferences. *Comput. Intelligence* 20(2): 111–136.

Dyer JS, Fishburn PC, Steuer RE, Wallenius J, Zionts S (1992) Multiple criteria decision making, multiattribute utility theory: The next ten years. *Management Sci.* 38(5):645–654.

Embrechts P, Kluppelberg C, Mikosch T (1997) *Modelling Extremal Events for Insurance and Finance* (Springer, New York).

Farias VF, Jagabathula S, Shah D (2013) A nonparametric approach to modeling choice with limited data. *Management Sci.* 59(2):305–322.

Furnkranz J, Hullermeier E (2011) *Preference Learning* (Springer, New York).

Green P, Srinivasan V (1978) Conjoint analysis in consumer research: Issues and outlook. *Consumer Res.* 5(2):103–123.

Green P, Srinivasan V (1990) Conjoint analysis in marketing: New developments with implications for research and practice. *J. Marketing* 54(4):3–19.

Hullermeier E, Furnkranz J, Cheng W, Brinker K (2008) Label ranking by learning pairwise preferences. *Artificial Intelligence* 172:1897–1916.

Kahneman D, Tversky A (1979) Prospect theory: An analysis of decision under risk. *Econometrica* 47(2):263–291.

Krzemienowski A (2009) Risk preference modeling with conditional average: An application to portfolio optimization. *Ann. Oper. Res.* 165:67–95.

Lovász L, Vempala S (2006) Hit-and-run from a corner. *SIAM J. Comput.* 35(4):985–1005.

Miettinen K, Ruiz F, Wierzbicki AP (2008) Introduction to multiobjective optimization: Interactive approaches. Branke J, Deb K, Miettinen K, Slowiński R, eds. *Multiobjective Optimization: Interactive and Evolutionary Approaches* (Springer-Verlag, Berlin), 27–57.

Ogryczak W (1998) Stochastic dominance relation and linear risk measures. Skulimowski AMJ, ed. *Financial Modeling—Proc. 23rd Meeting of the EURO Working Group on Financial Modeling* (Progress and Business Publishers, Warsaw), 191–212.

Ogryczak W (2014) Tail mean and related robust solution concepts. *Internat. J. Systems Sci.* 45(1):29–38.

Rockafellar RT, Uryasev S (2000) Optimization of conditional value-at-risk. *J. Risk* 2:21–41.

Roy B (1998) A missing link in operational research decision aiding: Robustness analysis. *Foundations Comput. Decision Sci.* 23:141–160.

Smith RL (1984) Efficient Monte Carlo procedures for generating points uniformly distributed over bounded regions. *Oper. Res.* 32(6):1296–1308.

Toubia O, Hauser JR, Garcia R (2007) Probabilistic polyhedral methods for adaptive choice-based conjoint analysis: Theory and application. *Marketing Sci.* 26(5):596–610.

Toubia O, Hauser JR, Simester DI (2004) Polyhedral methods for adaptive choice-based conjoint analysis. *J. Marketing Res.* 41(1):116–131.

Toubia O, Hauser JR, Simester DI, Dahan E (2003) Fast polyhedral conjoint estimation. *Marketing Sci.* 22(3):274–303.

Tversky A, Kahneman D (1991) Loss aversion in riskless choice: A reference-dependent model. *Quart. J. Econom.* 106(4):1039–1061.

Vempala S (2005) Geometric random walks: A survey. *Comput. Geometry* 52:573–612.

von Neumann J, Morgenstern O (1944) *Theory of Games and Economic Behavior* (Princeton University Press, Princeton, NJ).

Wald A (1945) Statistical decision functions which minimize the maximum risk. *Ann. Math.* 46:265–280.

Wallenius J, Dyer JS, Fishburn PC, Steuer RE, Zionts S, Deb K (2008) Multiple criteria decision making, multiattribute utility theory: Recent accomplishments and what lies ahead. *Management Sci.* 54(7):1336–1349.

Wang JG, Zionts S (2006) The aspiration level interactive method (AIM) reconsidered: Robustness of solutions to multiple criteria problems. *Eur. J. Oper. Res.* 17(2):948–958.

Zionts S, Wallenius J (1976) An interactive programming method for solving the multiple criteria problem. *Management Sci.* 22(6):652–663.

Dimitris Bertsimas is the Boeing Professor of Operations Research, Codirector of the Operations Research Center at Massachusetts Institute of Technology, and a member of the National Academy of Engineering. He works on optimization, applied probability, and data mining and their applications. The paper in this issue is part of the author's research in the context of optimization.

Allison O'Hair is a lecturer of operations research and statistics at the MIT Sloan School of Management. Her research interests lie in the fields of optimization, analytics, and healthcare applications. The paper in this issue is part of the author's research in the context of optimization.