Advice Provision in Multiple Prospect Selection Problems

Amos Azaria[1] and Sarit Kraus[1,2]

[1] Department of Computer Science, Bar-Ilan University, Ramat Gan 52900, Israel
[2] Dept. of Industrial Engineering Jerusalem College of Technology, Israel

Abstract. When humans face a broad spectrum of topics, where each topic consists of several options, they usually make a decision on each topic separately. Usually, a person will perform better by making a global decision, however, taking all consequences into account is extremely difficult. We present a novel computational method for advice-generation in an environment where people need to decide among multiple selection problems. This method is based on the prospect theory and uses machine learning techniques. We graphically present this advice to the users and compare it with an advice which encourages the users to always select the option with a higher expected outcome. We show that our method outperforms the expected outcome approach in terms of user happiness and satisfaction.

1 Introduction

It is hard to overestimate the importance of decision-making in life. People make decisions on a daily basis; some are trivial such as whether or not to eat ice-cream, while others are important such as which apartment to buy. "Choice Bracketing", termed by Read et al. [14], designates the grouping of individual choices together into sets. "Broadly Bracketing" indicates that the decision-maker takes all choices into account when making his decision, while "Narrow Bracketing" indicates that the decision-maker isolates each choice from all other choices. Of course, in order to bracket several decisions, they must all be converted to a single scale (such as money), where a decision that turns out to be a bad choice for one topic may be balanced out by another decision which turns out to be a good choice for a different topic.

It has been shown that people tend to use narrow bracketing and usually treat each decision as if in isolation from all other decisions, which in many cases results in a poor choice [20,14,15,3]. Gneezy and Potters [6] have shown that, in investment games, people who are forced into broadly bracketing by viewing their revenue less often become less risk averse and therefore increase their average revenue (this finding has been replicated by Moher and Koehler [11], however they have failed to replicate it in generalized settings). Unfortunately, these studies did not test users' satisfaction, only their average total revenue.

A classic experiment that illustrates narrow bracketing was done by Tversky and Kahneman [20]. They asked their subjects the following question:
"Imagine that you face the following pair of concurrent decisions. First examine both decisions, then indicate the options you prefer:

Choice I. Choose between:
A. A guaranteed gain of $\$240$.
B. A 25% chance to gain $\$1000$ and a 75% chance to gain nothing.
Choice II. Choose between:
C. A guaranteed loss of $\$750$.
D. A 75% chance to lose $\$1000$ and a 25% chance to lose nothing."

Since people tend to be risk averse with a positive payoff and risk seeking with a negative payoff, a large majority of subjects (73%) chose both A and D. Only 3% of the subjects chose B and C. Combining A and D yields a 25% chance to gain $\$240$ and a 75% chance to lose $\$760$. However, combining B and C dominates this with a 25% chance to gain $\$250$ and a 75% chance to lose $\$750$. Indeed, when presenting all four combined choices (A+C, A+D, B+C and B+D), no one chose the dominated option (A+D). This experiment shows that although people tend to narrow their bracketing, they make better choices when explicitly broadening it.

We tackle people's tendency to narrow bracketing using an environment where people need to decide among several independent selection problems, whether they prefer a guaranteed outcome or a higher, but uncertain, outcome. We build an agent which first learns peoples' preferences and generates a general human model. The agent then searches the space of all possible combinations that can be chosen by the user and, based on the human model, advises the user which combination would be most beneficial for him.

Many real life situations resemble our problem. The most obvious example is when building an investment portfolio, where some stocks have a higher risk but also offer an opportunity to receive higher interest. On the other hand one can invest in a bond with a lower risk and a lower interest level. Most people combine different stocks and bonds, combining different levels of risk. A similar example might be a manager who wants to market several products, where each product has its own probability for success and failure and its estimated revenue. Our agent may be integrated into a Decision Support System which manages use on regular basis.

Another example might be a GPS guiding a driver who can choose between a longer road with a low probability of heavy traffic and a shorter road with a high probability for heavy traffic. If this decision is made several times on a trip the time consumption associated with choosing a road which turned out to be with heavy traffic may be canceled by future decisions in which the traffic was flowing. Therefore, considering the full route may result in greater satisfaction for the driver than making a separate decision for each and every junction.

For the last example suppose that a professor has several papers that she would like to submit to conferences. When considering each paper separately, the professor may send each paper to a conference to which it is likely to be accepted, however, when considering all papers together, the professor may prefer sending some papers to a highly refereed conference where it is likely that at least a few of them are accepted.

We would like to emphasize, that although we focus on advice provision, the agent may utterly replace the decision-maker. For instance, if the GPS mentioned above is embedded into an autonomous vehicle, it may take the human to his target without asking the human which route to take.

The main contributions of the paper are the following: first, we present the algorithmic challenge that is associated with the human bracketing problem. Second, we developed a multistage procedure for providing humans with a combination of choices in multiple prospect selections aiming at maximizing human satisfaction. The innovation in the preference elicitation of the multistage procedure stems from the use of an off-line learning group. New users do not need to provide any additional information and can receive advice with no learning period. Furthermore, the group elicitation procedure is done using simple problems in which humans can provide their real preferences and address the inability of people to recognize their preferences in complex decision problems. At last, we provide extensive experiments that show the success of our proposed method in terms of human satisfaction.

2 Related Work

Recommendation systems use models for predicting user ratings in order to supply advice. (See Ricci et al. [16] for a recent review). The main application of recommender systems is to find an item (or items) among hundreds or thousands of items which the system expects would be favorable to the user based on his previous preferences. A common case would be a recommendation system which recommends a book or a movie to a user, based on movies or books that the user enjoyed in the past [7].

In a recent work, Azaria et al. [2] propose a method for giving advice to users in an environment where users have several options from which to choose where the decisions are made in sequential order. However, in their work they assume that the users and the agent have different goals and utility functions, while in our work the agent's goal is solely to help the user and thus they both have identical utility functions.

Sarne et al. [17] attempt to facilitate people's decision-making process by modifying the presentation of the problem in an economic search environment. However, in their work they assume that people want to maximize their expected monetary value. We relax this assumption and assume that people have a non-monetary utility function which they try to maximize. In this paper we base this utility function on the Prospect Theory.

3 Prospect Theory

The Prospect Theory was presented by Kahneman and Tversky in [10] and later refined to the Cumulative Prospect Theory in [21]. The Prospect Theory is based on three principles. The first is that people do not take into account their total wealth when accepting or rejecting an uncertain opportunity (as suggested by the expected utility hypothesis [5]), but rather use their current wealth as a baseline and will be happy if they win an amount and become upset if they lose an amount. The second principle is loss aversion, where people hate losing more than they like winning. The third principle is that people have a subjective representation of probabilities and do not interpret probabilities fully rationally, but rather use their own decision weights when deciding whether to reject or accept a gamble. In his book, Kahneman [9] (p.314) gives the following examples: The decision weight that corresponds to a 90% chance is 71.2%, while the decision weight

that corresponds to a 10% chance is 18.6%. According to these examples, people are likely to prefer a guaranteed outcome of $\$80$ than a gamble with a 90% chance of winning $\$100$, since the latter is only worth $\$71.2$ to them. Tversky and Kahneman elicited these weights by sequentially asking subjects to choose between a specific lottery and many different guaranteed outcomes. The equivalent to the given lottery for a certain subject was set to the average between the greatest rejected guaranteed outcome and the smallest accepted guaranteed outcome. However, these decision weights depend on peoples personality, their wealth, culture and the scope of the payoff in question. We build a human model by learning the decision weights which best suit our population and scope of payoff based on machine learning techniques.

4 Multiple Prospect Selections Advice Provision

The term *prospect* comes from Kahneman and Tversky, who refer to a lottery where a player has a chance to win (or lose) a certain amount as a prospect. We only use *simple prospects*, where a player can gain a certain outcome with some probability, and zero otherwise. A *prospect selection problem* $s =< x; p, y >$, is a problem where a player needs to choose between a guaranteed outcome of x and a probability of p to win an outcome y (we use the subscripts s_x, s_p and s_y respectively). We use $c \in \{g, u\}$ to denote the user's choice, where g denotes a choice of the guaranteed value (x), and u denotes the uncertain outcome (the prospect: y with probability p). An important point is that a fully rational player (i.e. one who maximizes expected monetary outcome) would always choose the uncertain outcome if $p \cdot y \geq x$, and the guaranteed outcome otherwise. However, as shown by Tversky and Kahneman [21], people do not act fully rationally not because they cannot do so, but because they use different decision weights.

In our work we consider *multiple prospect selection problems* where a player faces k (different) prospect selection problems. As mentioned in the introduction, people fail to broaden their bracketing and treat each selection problem as if it is isolated from the other selection problems. Therefore, people use their original decision weights, which are sub-optimal for multiple selection problems.

Our goal is to build an agent that advises a human player in which of the selection problems to choose the guaranteed outcome and in which to choose the prospect. We denote this advice by $a \in \{g, u\}$. We measure the performance of the agent in terms of human satisfaction from the final result, human satisfaction from the decisions he made, human satisfaction from the advice and the fraction of selection problems where the human followed the advice. We intentionally do not measure success in terms of the raw final stake, since, as mentioned above, people care about how much they have only in context, that is, in comparison to how much they could have had (or lost), and what risks they took or avoided in order to achieve that outcome.

We now present the prospect Selection problem Advice provider for Multiple Problems agent (SAMP). To build this agent we first use machine learning to elicit decision weights from given data. Based on these decision weights we build a general human model that can calculate the utility in human eyes for any combination of choices in a multiple prospect selection problem. Upon demand, SAMP searches for the best combination using this model and presents it to the user.

4.1 Decision Weights Elicitation

The first step in building SAMP, is eliciting human decision weights. Since these decision weights will be used to build a *general* human model, we collect data using subjects from a similar culture and using a similar scope of outcome and probabilities. However, we do not collect personal parameters regarding users' preferences (such as risk attitude). This data is collected from subjects who were asked to choose between a guaranteed outcome and a simple prospect. Note that this is a very simple choice and we therefore assume that humans can provide their real preferences.

Using the data we build a logistic regression classifier. We feed the classifier for each prospect selection problem with the probability of winning in the prospect, the ratio between the prospect outcome and the guaranteed outcome, and all quadratic combinations of the two. The classifier needs to classify the data based on the subjects' choices. Adding quadratic combinations is required since a linear combination of the two features isn't enough to learn a good model. Quadratic functions are used with success as SVM kernels [19]. More formally, the feature vector for the classifier is:

$$v = \{s_p,\ \frac{s_y}{s_x},\ {s_p}^2,\ (\frac{s_y}{s_x})^2,\ s_p \cdot \frac{s_y}{s_x}\}$$

and the classifier is trained on c. We use $\frac{s_y}{s_x}$ rather than s_y because we want the final decision weights learned to be independent of the outcome and solely depend on the probability.

Given a new prospect selection problem feature vector the classifier will output a number between 0 and 1 which, if greater than 0.5, indicates that the user is more likely to choose one option and if smaller than 0.5 indicates that the user is more likely to choose the other. However, since we are interested in learning the decision weights, we would like to know when people are *indifferent* between choosing the guaranteed and the uncertain outcome. We therefore are interested in the cases where the classifier will output exactly 0.5. Since we use a logistic regression classifier, we obtain:

$$\frac{1}{1+e^{-(w^T \cdot v)}} = 0.5 \tag{1}$$

where w is the vector of weights obtained from the classifier and v is the feature vector described above. This implies: $1 + e^{-(w^T \cdot v)} = 2$, and therefore: $e^{-(w^T \cdot v)} = 1$, which implies: $(w^T \cdot v) = 0$. Since we are interested in finding the decision weights for a given probability, we denote: $z = \frac{s_y}{s_x}$ and solve Equation 4.1 for z. Writing both the feature vector and the weight vector explicitly we obtain:

$$w_0 + w_1 \cdot s_p + w_2 \cdot z + w_3 \cdot s_p + w_4 \cdot z^2 + w_5 \cdot s_p \cdot z = 0 \tag{2}$$

Solving Equation 2 we obtain:

$$z(s_p) = \frac{-w_5 - w_2 \pm \sqrt{(w_5 s_p + w_2)^2 - 4w_4(w_3 {s_p}^2 + w_1 s_p + w_0)}}{2w_4} \tag{3}$$

Given a probability s_p, the actual decision weight $d(s_p)$ is $\frac{1}{z(s_p)}$. Only a single solution is appropriate; in most cases the second solution is either negative or greater than 1. We set $d(1)$ to 1 and $d(0)$ to 0. If in any other case $d(s_p) > 1$ or $d(s_p) < 0$, we set it to 1 or 0 respectively (this should not happen if the classifier was trained on sufficient data).

4.2 Assessing the value of multiple prospects

Using the decision weights obtained above, we can assess a value of a prospect. Given a prospect with a probability s_p of winning an outcome s_y, using the decision weight $d(s_p)$ we obtain that the value of the prospect for the user is simply:

$$u(s_p, s_y) = d(s_p) \cdot s_y \tag{4}$$

The main challenge which remains is to assess the value of multiple (k) prospects. If we were simply to sum the value of all prospects individually, we would fail by using the exact concept which we are trying to overcome, i.e. narrow bracketing. The first step in assessing the value of multiple prospects is calculating the final probability for each of the possible outcomes. Note that there may be up to $n = 2^k$ possible outcomes. These outcomes must be sorted from low to high. We use the following algorithm to efficiently (linear in output) obtain all possible outcomes (along with their probabilities):

Input: Pml - A list of selection problems ($s =< x; p, y >$) and user's selections on these problems ($c \in \{g, u\}$).

Output: OP - A hushmap holding all possible outcomes as keys and their probabilities as values.

1: $OP[0] \leftarrow 1$
2: **for** each problem s in Pml **do**
3: **if** guaranteed scenario was selected ($c = g$) **then**
4: **for** each $outcome$ in OP **do**
5: increase $outcome$ by guaranteed value (s_x)
6: **else**
7: $Temp \leftarrow OP$
8: clear OP
9: **for** each $outcome$ in OP **do**
10: $OP[outcome] \leftarrow Temp[outcome] \cdot (1 - s_p)$ [1]
11: $OP[outcome + s_y] \leftarrow Temp[outcome] \cdot s_p$
12: return OP

Once we obtain the probabilities $\boldsymbol{p} = p^1, p^2 \dots p^n$ ($\sum_{i=1}^{n} p^i = 1$) and outcomes $\boldsymbol{y} = y^1, y^2 \dots y^n$ ($y^1 < \dots < y^n$), based upon the Cumulative Prospect Theory [21], we assess the value for the user by:

$$u(\boldsymbol{p}, \boldsymbol{y}) = \sum_{i=1}^{n} d\big(\sum_{j \geq i} p^j\big) y^i - d\big(\sum_{j > i} p^j\big) y^i \tag{5}$$

We now explain the intuition behind this method. Note that when $i = 1$, $\sum_{j \geq 1} p^j = 1$ and since $d(1) = 1$, then $d\big(\sum_{j \geq 1} p^j\big) y^1$ is simply y^1. This illustrates the fact that the user is guaranteed an outcome of at least y^1 (since y^1 is the lowest possible outcome). The second position's contribution ($i = 2$) illustrates the fact that the user has a probability of $1 - p^1$ to obtain at least y^2. The second term in the sum omits

[1] In 10 and 11, if $OP[outcome]$ or $OP[outcome + s_y]$ already have a value, increment that value by $Temp[outcome] \cdot (1 - s_p)$ or $Temp[outcome] \cdot s_p$ accordingly.

multiple counting of the same outcome. We now show that Equation 5 generalizes a single prospect. Given a single prospect, there are two possible outcomes $\boldsymbol{p} = 0, s_y$, and the probabilities are $\boldsymbol{y} = 1 - s_p, s_p$. Assigning p and y in Equation 5 we obtain: $u(\boldsymbol{p}, \boldsymbol{y}) = d(1) \cdot 0 - d(s_p) \cdot 0 + d(s_p) \cdot s_y - 0 \cdot s_y$ which is identical to Equation 4.

In order to be more efficient, in practice we do not sum the probabilities in every iteration but start with 1 and subtract the current probability in every iteration.

4.3 Advice Provision

Given a Multiple Prospect Selection Problem, each combination of choices yields different vectors $\boldsymbol{p}$ and $\boldsymbol{y}$ and therefore using the model above, each combination of choices yields a different user value ($u(\boldsymbol{p}, \boldsymbol{y})$). SAMP searches for a combination of choices which maximizes this value. This search can be invoked using any search method (such as genetic algorithm or interior-point).

4.4 Fully Rational Agent

Since human full rationality assumption is broadly used and is a very common assumption, we compare the performance of SAMP to the performance of a *rational agent* which assumes human full rationality (i.e. assumes that people want to maximize their expected monetary value). Performance is measured in terms of human satisfaction. In all selection problems the rational agent advises the user to choose the uncertain outcome if $s_p \cdot s_y \geq s_x$, and the guaranteed outcome otherwise. The rational agent's advice does not depend on the number of selection problems k. A user following the rational agent's advice, will maximize his expected outcome.

5 Experimental Design

We ran our experiments using Amazon's Mechanical Turk (AMT) [1]. AMT has become an important tool for running experiments and was established as a viable method for data collection [12].

We recruited 52 participants for SAMP's learning phase and 202 participants for evaluating both SAMP and the fully rational agent. Two subjects were removed from the evaluation process due to invoking a response in less than 15 seconds, which was considered unreasonably fast, and we suspected that they just wanted to move on to their next task. All other subjects took at least 25 seconds to submit their preferences, with an average of 94.7 seconds.

In the evaluation phase, each subject received a single advice, either from SAMP or from the fully rational agent. In the learning phase the subjects did not receive any advice. Each subject participated only once. 52% of the subjects were males and 48% were females. Subjects' ages ranged from 18 to 75, with a mean of 32.2 and median of 29. All subjects were residents of the USA.

We set $k = 5$, i.e. the subjects had to make their choice regarding five prospect selection problems. The guaranteed outcome was drawn uniformly between 2 and 10 cents, and the probability to win the uncertain outcome (s_p) was drawn uniformly from

$\{0.05, 0.1, 0.15, 0.2, ..., 0.95\}$. Choosing the uncertain outcome yield upto 35% more on the expected utility value of the guaranteed outcome.

Once the subjects selected their preferences, we invoked a lottery on each of the prospect selection problems and paid them according to their preferences. I.e. if a subject chose the guaranteed outcome for a certain selection problem he was paid accordingly, and if he chose the uncertain outcome, the system randomly generated a number r in $[0, 1]$ and paid the subject s_y if $r \leq s_p$. In order to encourage narrow bracketing in the learning phase, the subjects were told to answer each selection problem as if it were standalone, and instead of having the system invoke all the selection problems, the system randomly selected a single selection problem and invoked only that one. The subjects were fully aware of this process. In the evaluation phase, however, the system obviously invoked all five prospect selection problems.

In the evaluation phase, in addition to the five selection problems, the subjects were also presented with the agent's advice. Prior to receiving the actual advice, the subjects were told that the advice is provided by a third party agent which is trying to help them. The subjects were shown the consequences of following the advice using a pie chart, which indicated the actual probability of winning each possible outcome. A pie chart is a common method of showing how a full resource is split up. Therefore, in our experiments, we use the pie chart to show how the certainty is split to smaller probabilities. This method enables easy comprehension and provides all information in a single chart.

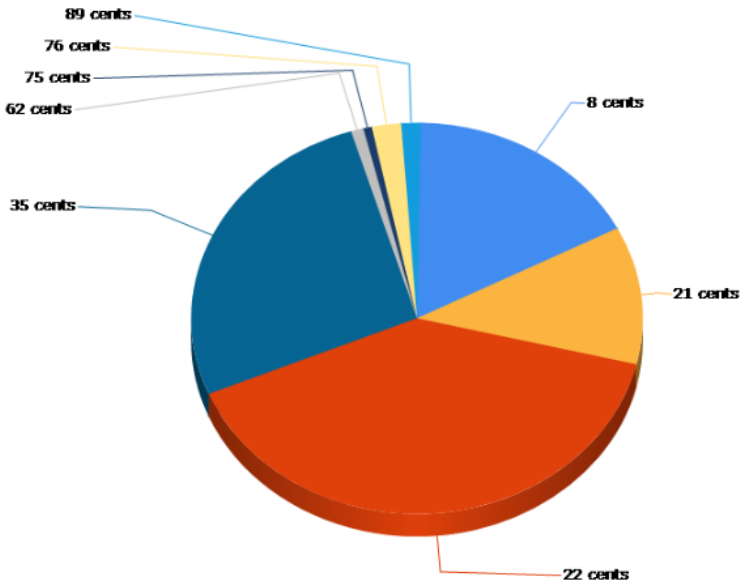

Fig. 1. A visualization of the agent's advice

Figure 1 presents an example for a visualization of the agent's advice. In this example, if the user follows the exact advice given by the agent, he is guaranteed a win of at least 8 cents, however, he might win upto 89 cents. The interpretation of the pie chart was explained to the subjects, and their comments indicated that they clearly understood this interpretation. After submitting their choices and told the final results, the subjects were asked the following three questions:

1. Are you happy with the final result?
2. Are you happy with the decisions you made?
3. How good was the advice given to you by the system?

The subjects answered these questions on a 1 to 5 scale, where 1 is associated with "not at all" in the first two questions, and "very bad advice" in the third question, 3 is associated with "o.k.", and 5 is associated with "very happy" in the first two questions, and "very good advice" in the third question. All subjects were also asked to give comments if they had any.

5.1 SAMP construction

Recall that in SAMP construction, after the data is collected a logistic regression classifier is built using this data. This classifier is built in order to find the decision weights, therefore the classifier is trained on all the data. However, in order to evaluate the performance of the classifier we also ran a 10 fold cross validation on the data. We found that the accuracy of the classifier was 67.2% (F-Measure 0.67). These results are satisfying, as we are considering a general human model and people differ in their risk attitudes. Nonetheless, this classifier allows us to build a good enough model for people's decision weights. Using other classifiers yield similar results (e.g. SVM yields accuracy of 67% and neural networks yield 65.6%), however, we use logistic regression so we can extract the model as described in section 4.1.

After extracting the parameters we were able to calculate the decision weights for any given probability. Several decision weights learned by SAMP for some selected probabilities can be found in Table 1. For example the decision weight associated with 0.25 is 0.20. This indicates that people should be indifferent between a prospect that offers a 25% chance to win 10 cents and a guaranteed outcome of 2 cents. Note that the decision weights are lower than the actual probabilities in all cases; this indicates that, in our environment, people were risk averse regardless of the probability.

Probability	Decision Weight	Probability	Decision Weight
0.10	0.07	0.60	0.48
0.25	0.20	0.75	0.64
0.40	0.32	0.90	0.85

Table 1. Decision weights for selected probabilities

Using these decision weights SAMP builds a general human model as described in section 4.2. Since we used a small k, we had a small search space and therefore

SAMP used an exhaustive search when searching for the best combination to be advised to the users. Figure 1 presents an example of a SAMP's advice, in a scenario where the guaranteed outcome among all selection problems sums up to 21 cents. Following the advice seems preferable to choosing the guaranteed outcome, as the probability of gaining less than the guaranteed outcome by following the advice is less than 14% and the user is very likely to gain more than the guaranteed outcome, with a slight chance of gaining much more. The expected utility from following SAMP's advice is 25.7 cents, however, SAMP evaluated the human value of its advice in 23.4 cents (greater than any other combination of choices).

5.2 Results

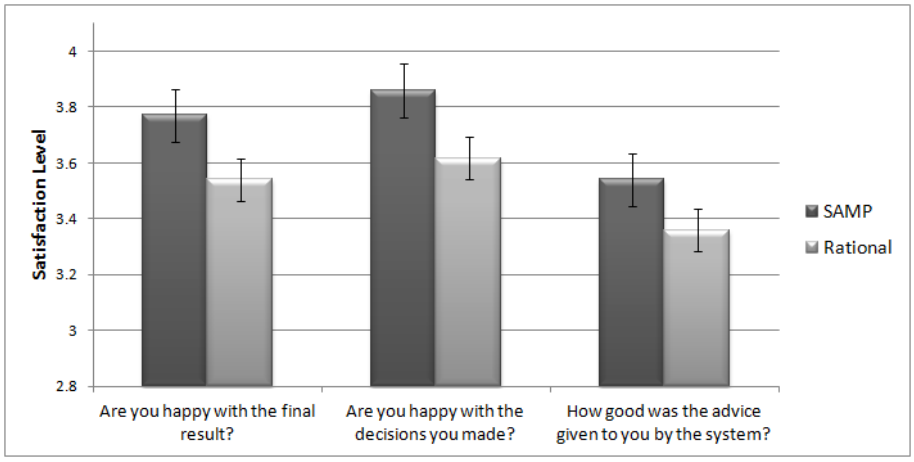

Fig. 2. Comparison between user satisfaction levels among subjects who received SAMP's advice and those who received the fully rational agent's advice

Figure 2 presents the final user satisfaction levels for users who received SAMP's advice and users who received the fully rational advice. The higher the satisfaction level the better the result. As can be seen in the figure, in all three questions SAMP outperformed the rational agent. ANOVA test on all three parameters (together) indicates that these results differ significantly ($p < 0.05$). In the first two questions results differ significantly ($p < 0.05$) also using the student's t-test between the two groups, however, the results obtained from the third question did not reach significance level by the student's t-test (although they were very close to achieving it, with $p = 0.07$).

We also compare the average fraction of advice followers, i.e. the fraction of prospect selection problems where each user followed the advice for that selection problem. The advice given by SAMP for each of the selection problems was followed 76.5% of the time, while the advice given by the rational agent was followed only 70.8% of the time. Since more users followed the advice given by SAMP, on average, users who received

SAMP's advice resulted with a non-statistically significant higher outcome (34.5) than those who received the rational advice (32.2). However, we attach minor importance to this achievement since SAMP is not destined to increase the users' expected outcome but to increase their satisfaction.

To sum up the results, the subjects' satisfaction level in all parameters was greater when they received SAMP's advice than when they received the fully rational agent's advice. This indicates that the assumptions used by SAMP are more accurate than assuming full rationality. However, the rational agent also performed well, since its advice was followed in a large majority of selection problems. The rational agent also reached an average score of 3.36 when asked how good it was (recall that 3 is associated with "o.k."). Therefore, the rational agent may be good enough for a cold start (when no training data is available).

Many subjects appreciated the pie chart display as we received comments that praised it. One example is: "The pie chart made it much easier for me to make my decision."

6 Discussion and Future Work

Research has shown that people act differently when the stakes are higher, and tend to be more risk-averse [8]. Nonetheless, many users commented that they would act differently if the stakes were higher. Since people tend to reject favorable prospects when the stakes are higher, narrow bracketing will cause them to reject many more favorable prospects than they should. We therefore expect that, when the stakes are higher, our agent will have a larger impact on the user satisfaction. However, in our settings raising the stakes would not allow us to actually pay the subjects and the prospect selection problems would have needed to be hypothetical (for a comparison between hypothetical and real scenarios see [13]). Another option would be to retain the current stakes but conduct the research in a different culture where the average monthly expenditure of participants is much lower. Using a different culture instead of actually raising the stakes is a common method used in psychological experiments [8,4].

In many occasions the topics involved may be dependent, where a certain result in one topic may increase or decrease the probability for a different result in a second topic. For example, if there is heavy traffic on one road, this may be a sign that there will also be heavy traffic on a second road. In such cases SAMP may easily be generalized using Bayesian inference. However, since humans encounter great difficulty in calculating Bayes' rule themselves (unless properly trained [18]), therefore, SAMP may be even more of an assist in such cases.

It remains questionable, what will happen as the number of prospect selection problems increases (k). In such a case, the pie chart may become very complex as the number of outcome options increases exponentially. One option is to group up similar outcomes, but still it is unclear how to do so.

In current work, the experiments were based on a single trial. We intend to test a scenario where the user doesn't see all of the selection problems at once, but must make his selections sequentially. This scenario is quite common in real life where one usually faces sequential decisions. In such a scenario, the user (and therefore the agent) has

only partial data on future selection problems, making it harder to advise the user as to which option to choose for each selection problem. The user on the other hand, might come to appreciate the value of the advice better over time and increase his trust in it. The agent may also gain trust by informing the user what he may have won had he followed the advice.

In future work we also intend to build a personalized agent that will provide different advice depending on the user with whom it is interacting. We will either need to find a good method to cluster the training data into different clusters and have SAMP provide advice depending on the user's cluster, or completely personalize the agent using parameters that could be different for every user. When interacting with a user, the agent will first collect data on the specific user and then provide advice which is best suited for him. One way to collect data is by asking the user to first make his decisions as if each selection problem is a standalone and then the agent will advise the combination which is best for that specific user. We expect this method to significantly improve the agent's performance in terms of user satisfaction. However, this method would require a longer procedure for data collection, and the users participating in the learning phase would be required to answer an additional questionnaire. A personalized agent will also require learning data from more subjects.

7 Conclusions

When people need to make decisions regarding several topics they tend to view each topic as if it were on its own and when deciding which action to take they ignore all other topics. This tendency was shown to be harmful, and people would perform better if they could make a more intelligent selection based on all topics together. Helping people in such cases isn't simple since people do not necessarily want to maximize their expected monetary value.

In this paper we present an agent which advises people in an environment which includes multiple prospect selection problems, where in each selection problem the user must choose between a guaranteed outcome and an uncertain outcome. This agent collects data on humans in the desirable environment and, based on the data it builds a general human model using prospect theory concepts. Using this model, the agent searches for a combination of selections which is most favorable for humans and recommends it to the user. We present the resulted combination using a pie chart which visualizes the probability of each possible outcome. The advice composed by our agent significantly outperforms fully rational advice (i.e. advice which maximizes expected utility) in terms of user satisfaction.

Reaching such an achievement using a general human model is conspicuous since people differ from each other, and many times advice which might be good for one user may not be as good for another. In this work we have proven the concept of using decision weights, showed how to learn them from data and generalize their use to a combination of multiple prospect selection problems.

8 Acknowledgment

This work was supported in part by ERC grant $\#267523$, the Google Inter-university center for Electronic Markets and Auctions, MURI grant number $W911NF-08-1-0144$ and ARO grants $W911NF0910206$ and $W911NF1110344$.

References

1. Amazon. Mechanical Turk services. http://www.mturk.com/, 2012.
2. A. Azaria, Z. Rabinovich, S. Kraus, C. V. Goldman, and Y. Gal. Strategic advice provision in repeated human-agent interactions. In *AAAI*, 2012.
3. Nicholas Barberis, Ming Huang, and Richard H. Thaler. Individual preferences, monetary gambles, and stock market participation: A case for narrow framing. *American Economic Review*, 96(4):1069–1090, 2006.
4. L. A. Cameron. Raising the stakes in the ultimatum game: Experimental evidence from Indonesia. *Economic Inquiry*, 37(1):47–59, 1999.
5. M. Friedman and L. J. Savage. The expected-utility hypothesis and the measurability of utility. *The Journal of Political Economy*, 60(6):463–490, 1952.
6. Uri Gneezy and Jan Potters. An experiment on risk taking and evaluation periods. *Quarterly Journal of Economics*, pages 631–645, 1997.
7. Jonathan L. Herlocker, Joseph A. Konstan, Loren G. Terveen, and John T. Riedl. Evaluating collaborative filtering recommender systems. *ACM Trans. Inf. Syst.*, 22(1):5–53, January 2004.
8. Steven J Kachelmeier and Mohamed Shehata. Examining risk preferences under high monetary incentives: Experimental evidence from the people's republic of china. *American Economic Review*, 82(5):1120–41, December 1992.
9. Daniel Kahneman. *Thinking, fast and slow*. Allen Lane, London, 2011.
10. Daniel Kahneman and Amos Tversky. Prospect theory: An analysis of decision under risk. *Econometrica*, 47(2):263–291, 1979.
11. Ester Moher and Derek J. Koehler. Bracketing effects on risk tolerance: Generalizability and underlying mechanisms. *Judgment and Decision Making*, 5(5):339–346, August 2010.
12. G. Paolacci, J. Chandler, and P. G. Ipeirotis. Running experiments on Amazon Mechanical Turk. *Judgment and Decision Making*, 5(5), 2010.
13. Daniel Read. Monetary incentives, what are they good for? *Journal of Economic Methodology*, 12(2):265–276, 2005.
14. Daniel Read, George Loewenstein, and Matthew Rabin. Choice bracketing. *Journal of Risk and Uncertainty*, 19(1):171–197, December 1999.
15. Donald A. Redelmeier and Amos Tversky. On the framing of multiple prospects. *Psychological Science*, 3(3):191–193, May 1992.
16. F. Ricci, L. Rokach, B. Shapira, and P.B. Kantor, editors. *Recommender Systems Handbook*. Springer, 2011.
17. David Sarne, Avshalom Elmalech, Barbara J. Grosz, and Moti Geva. Less is more: restructuring decisions to improve agent search. In *Proc. of AAMAS*, pages 431–438, 2011.
18. J. Shanteau, M. Grier, J. Johnson, and E. Berner. Teaching decision-making skills to student nurses. *Teaching decision making to adolescents*, pages 185–206, 1991.
19. Thorsten Thies and Frank Weber. Optimal reduced-set vectors for support vector machines with a quadratic kernel. *Neural Comput.*, 16(9):1769–1777, September 2004.
20. A. Tversky and D. Kahneman. The framing of decisions and the psychology of choice. *Science*, 211(4481):453–458, January 1981.
21. Amos Tversky and Daniel Kahneman. Advances in prospect theory: Cumulative representation of uncertainty. *Journal of Risk and Uncertainty*, 5(4):297–323, October 1992.