Web Intelligence and Agent Systems: An International Journal 10 (2012) 385–395
DOI 10.3233/WIA-2012-0253
IOS Press

Minimization of decoy effects in recommender result sets

Erich Christian Teppan [a,*] and Alexander Felfernig [b]
[a] *Dept. of Applied Informatics, University of Klagenfurt, Universitaetsstrasse 65-67, 9020 Klagenfurt, Austria*
E-mail: erich.teppan@aau.at
[b] *Dept. of Softwaretechnology, Graz Instititute of Technology, Inffeldgasse 16b, 8010 Graz, Austria*
E-mail: alexander.felfernig@ist.tugraz.at

Abstract. Recommender systems are common web applications which support users in finding suitable products in large and/or complex product domains. Although state-of-the-art systems manage to accomplish the task of finding and presenting suitable products they show big deficits in their models of human behavior. Time limitations, cognitive capacities and willingness to cognitive effort bound rational decision making which can lead to unforeseen side effects and consequently to sub-optimal decisions. Decoy effects are cognitive phenomena which are omni-present on result pages but state-of-the-art recommender systems are completely unaware of such effects. Due to the fact that such effects constitute one source of irrational decisions their identification and, if necessary, the neutralization of their biasing potential is extremely important. This paper introduces an approach for identifying and minimizing decoy effects on recommender result pages. To support the suggested approach we present the results of a corresponding user study which clearly proves the concept. Moreover, this paper also investigates whether the decreasing impact of decoys on uncertainty levels during decision making is affected by the decoy minimization approach.

Keywords: Recommender systems, decision support systems, online decision making, decoy minimization

1. Introduction

Recommender systems [2,6–8,11,13] support the users of online sales platforms in finding and identifying items which best match their wishes and needs. Depending on the product domain different approaches have been shown to work well. Collaborative filtering [11,24] exploits user similarities (based on user profiles containing product/item preferences) in order to calculate recommendations, which are typically presented in the form of ‘Users who liked X also liked Y’. This approach is most suitable when the product domain is not complex (e.g. books) or when no additional product information is available for computing recommendations. Content-based recommendation [5,13] utilizes information about the recommended products (e.g. genre in the domain of movies) and matches it against a user profile. In addition to product information, knowledge-based recommenders (KBRs) [6] exploit deep domain knowledge (e.g. legislative restrictions). Furthermore, KBRs offer intelligent mechanisms for preference elicitation (e.g. interactive dialogs, tweaking critiquing interfaces), repair of inconsistent or contradicting requirements (stated by the user), or explanations as to why a certain product fulfills the user’s requirements [6,8]. Utility-based recommenders (UBRs) [2,6] constitute a special form of KBRs which calculate user-specific utilities of items by using pre-defined utility functions. On the result page of a recommender application, such items are then ordered compliant to their calculated utility.

What has been ignored so far in the design of decision support systems in general and UBRs in particular are psychological side effects which always occur when multiple items are presented concurrently such as on recommender result pages. Whereas the objective utility value of a certain item for a certain user (estimated by the utility function) is stable, the sub-

*Corresponding author. E-mail: erich.teppan@aau.at.

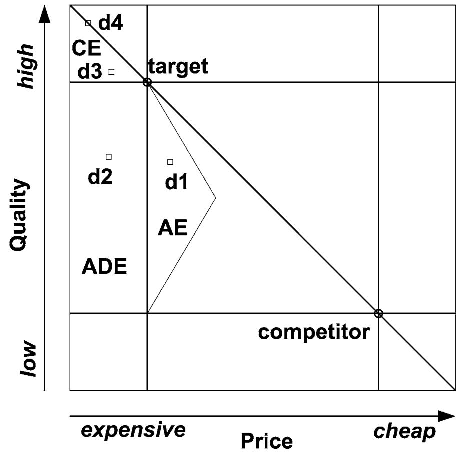


Fig. 1. Various decoys in a two-dimensional item landscape.

jective utility perceived by users is highly dependent on the surrounding items on the result page. This circumstance acts as a door opener for cognitive side effects which may lead to irrational decisions [12]. One big class of such influences are decoy effects [1,15,17–22]. A decoy effect is triggered when the addition of an additional item leads to an increased attraction (measured in terms of choice/selection/purchase) of another item already in the set. The item which is intended to increase another item's attraction is called decoy. The item(s) which is(are) intended to benefit is(are) called target(s). All other items are called competitors. Depending on the relative positions on the item landscape three types can be differentiated: Attraction effect (AE) [15,17], Asymmetric dominance effect (ADE) [1,20,22], and the Compromise effect (CE) [17,22]. Figure 1 shows the relative positions of example decoys producing the AE (d_1), ADE (d_2), or CE (d_3, d_4) in the two-dimensional case (i.e. only two product attributes, for example price and quality).

Initially there are only two concurrent items in the set: the target item, which is of higher quality but more expensive, and the competitor item which is of lower quality but cheaper than the target. The choice between the target and the competitor item (when presented to users) suffices a certain distribution function (e.g. 50% would choose the target and 50% would choose the competitor). When a decoy item (e.g. d_1, d_2, d_3, or d_4 in Fig. 1) is added to the choice set, the choice distribution changes for the benefit of the target item (e.g. 60% would choose the target). Different mechanisms are leading to the various decoy effects. When d_1 (AE) is presented the target item's attraction increases because the target item is much better than d_1 in terms of quality yet only little more expensive than d_1. This notion is also called tradeoff contrast [18]. The overall inferiority of d_1 compared to the target is more obvious than the overall inferiority compared to the competitor because d_1 is more similar to the target than to the competitor. d_2 (ADE) is totally dominated by the target (i.e. it is worse in both dimensions) but it is only worse in one dimension compared to the competitor. Additionally to the tradeoff contrasts d_2 is producing (ADE can be seen as the special case of AE), the asymmetric domination supports the perceived superiority of the target item. Tradeoff contrasts are also one mechanism which can be the reason for the CE. For example, d_3 is much more expensive than it is better in terms of quality than the target. A mechanism which is triggered additionally by d_3 and exclusively by d_4 is extremeness aversion [17,18]. People tend to choose a middle option rather than to take an extreme option. Adding d_3 or d_4 makes the target the intermediate option in both dimensions and therefore a good compromise between the decoy, the target, and the competitor. Please note that the diagrammatic description of decoy effects in Fig. 1 is not a decoy model and only serves the purpose of explanation. For a mathematical model of decoy effects in multi-dimensional item attribute spaces supporting any number of item set sizes see, for example [21]. Typically sets of items are presented to the user on recommender result pages and therefore items influence the attraction of each other. The question is not if decoy effects occur but rather how strong they are. If decoy effects are strong, users cannot rate the utility of the items objectively which may lead to suboptimal decisions.

The remainder of the paper is structured as follows: Section 2 gives an overview about work done in related research fields. Section 3 introduces a utility-based recommender approach for minimizing decoy effects on recommender result pages. Section 4 presents two user experiments about decoy-biased choice situations and proofs the concept presented in Section 3. Section 5 addresses the influence of decoys and decoy minimization on the uncertainty level during decision making. The paper is concluded in Section 6.

2. Related work

Many models of human behavior are grounded on the assumption that human decision making can be seen as fully rational [16]. In decision theory as well as in economical models the idea of humans as rational agents is wide spread. The notion of estimated value and estimated utility form the grounding of utility functions which still constitute the core of utility-based recommendation [14,16,25]. One of the most famous approaches for utility functions in utility-based recommender systems is multi attribute utility theory (MAUT) [14,25]. MAUT in its simplest form calculates the objective utility of an item for a specific user as the weighted sum over item attribute scorings. Item attribute scorings express how good an item performs in a particular attribute (like optical zoom, resolution, or price in the digicam domain). Typically scorings are in the range 1 (very bad) – 10 (excellent). The weights reflect the user's preferences and express how important this particular attribute is for the user (i.e. for one user the price of a digicam is most important whereas another user's focus lies on optical zoom). Against the ideal picture of the fully rational human the concept of bounded rationality assumes that human decision making cannot be seen as a fully rational process [12,23]. The results of empirical studies clearly show that people rather apply simpler heuristics than calculating a complete utility function [4,9,10]. Such heuristical simplification serves as the grounding for decision biases which can result in irrational product choice. Decoy effects are among the most well known effects responsible for the decision biases. These effects especially occur in situations where users have to select an item from a list of alternatives. Asymmetric Dominance-, Attraction- and Compromise Effects have been investigated in quite a lot of previous work [1,15,17–22]. Decision field theory [3] and prospect theory [12] represent two extensive models accounting for irrationalities in human choice behavior and offer the possibility of calculating subjective utility of an item. The simple dominance model [21] is an extension to MAUT and allows calculating the subjectively perceived utility of an item which is influenced by the surrounding set of presented items. In set-independent models like MAUT, the utility of an item for a user is calculated independently from other items. In set-dependent models like the simple dominance model, the utility of an item is rather calculated in terms of strengths and weaknesses compared to the other items in the set. For example, the same item can have a high utility when all other items are totally inferior, but a low utility if the other presented items are much better. The following Section shows how to combine set-independent utility functions (i.e. objective) with set-dependent utility functions (i.e. subjective) in order to identify decoy effects (i.e. biases which can lead to sub-optimal decisions) and furthermore how to neutralize/minimize those effects.

3. Concept for neutralizing decoys

3.1. Decoy effect identification

The first step to effectively minimize decoy effects on recommender result pages is to identify potential item set constellations which show high biasing potential. To this end, two types of utility models have to be applied in combination. First, a set-independent model (SIM) is needed to calculate objective utility values. In utility-based recommender systems such models are already applied for ordering the result items along their calculated utility (i.e. the utility function, e.g. MAUT). Second, a set-dependent model (SDM) is needed which calculates perceived item utilities depending on the surrounding item set. An example of an SDM for recommender systems is the Simple Dominance Model presented in [21]. The idea of combining those two model types is the following: Big differences between the set-independent utility (from SIM) and the set-dependent utility (from SDM) pinpoint to decoy effects. Basically, there can be distinguished three major constellations:

No differences between SIM and SDM. In this case there is no indication of a decoy effect. Figure 2 shows an example for this situation. Item A is the top-ranked item. The set-independent utility of each item is the same as the set-dependent utility.

Order-preserving differences. When there are differences between the SIM values and SDM values this indicates a potential decoy bias. The bigger the differences are the stronger is the indication for such effects. Figure 3 shows an example. Item A is again top-ranked by both models but the set-dependent utility of Item A is smaller than the set-independent utility. This means that in context of the presented item set the perceived utility of Item A is smaller than its real value.

Order-altering differences. This is the special case of the more general order-preserving case. Figure 4 shows an example. Here, the differences between the SIM values and the SDM values additionally result in a different value order. From the SIM point of view Item A is top-ranked but not from the SDM point of view.

The differentiation between the order-altering and order-preserving case has the following reason: Typically, different models operate on different scales, for

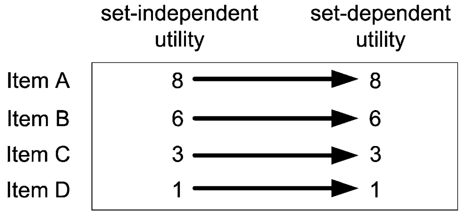


Fig. 2. No difference between SIM and SDM.

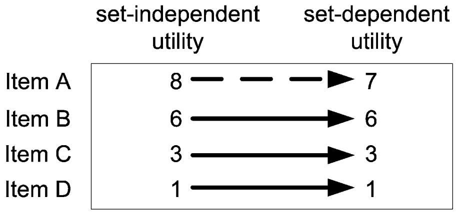


Fig. 3. Order-preserving differences between SIM and SDM.

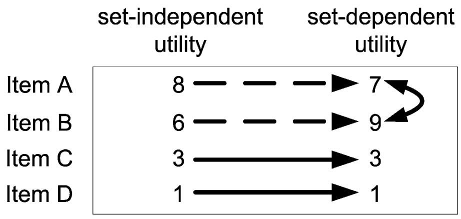


Fig. 4. Order-altering differences between SIM and SDM.

example one model might produce values between 0 and 100 whereas another one produces values between -1 and 1. One way of normalizing the different scales is to rank the items along the model utilities and compare the ranks instead of directly using the utility values. When using the item ranks only the order-altering example of Fig. 4 produces different ranks (compare Figs 5 and 6). Rank reordering is a strong indicator for decoy effects. By using item ranks instead of utility values we are loosing precision such that small decoy effects cannot be identified. What we are winning is model compatibility.

By comparing the item ranks produced by SIM and SDM we are now able to identify decoy effects. This forms the baseline for effectively neutralizing decoy effects which is discussed in the following sub-section.

3.2. *Decoy effect minimization*

Our approach of minimizing (=neutralizing) decoy effects is concentrating on getting the rank order produced by SDM in line with the rank order produced

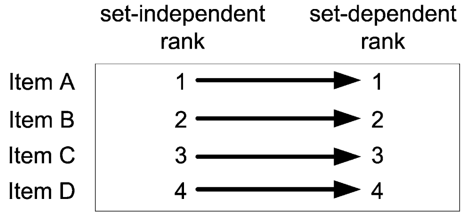


Fig. 5. Rank-preserving case between SIM and SDM.

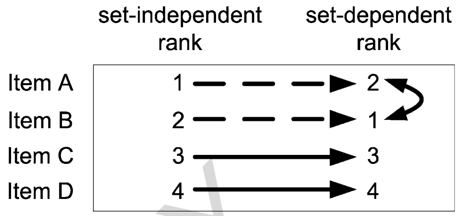


Fig. 6. Rank-altering case between SIM and SDM.

by SIM, rather than concentrating on the utility values directly. This has the following reasons:

1. Model normalization: The ranks produced by SIM and SDM are comparable.
2. Order-preserving decoy effects can be neglected: Also in context of the presented set, the top-ranked item seems to be the best one (for most users), the second-ranked item still seems to be the second best, and so on.
3. Complexity: To get continuous SDM utility values completely equal to SIM utility values is typically not possible. Discrete ranks are easier to handle.

The basis for our neutralization approach is that a certain subset (P) of a set of calculated solution items (S) is to be presented to the user. The calculation of S is done by the recommender system, by sorting out unsuitable items. This can be done by filters (knowledge-based) or by rejecting items below a certain utility threshold (utility-based). After that, the items are ranked along SIM and a set of top-ranked items T is identified depending on space limitations of the result page. In case that there are no decoy effects detected in the top-ranked items T (i.e. no order altering between SIM and SDM), the presented set P constitutes the top-ranked items (T). If there can be detected decoy effects in T there are two possibilities in order to minimize the biases. First, we can remove a subset $D \subset T$ causing the effects, such that $P = T - D$ (see Fig. 7). The second possibility for minimizing decoy effects is to add additional

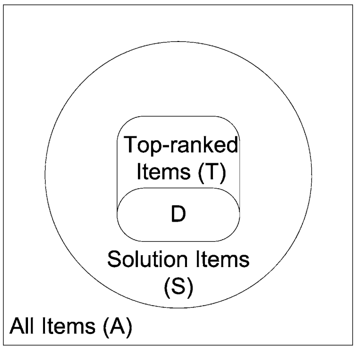


Fig. 7. Removing decoy elements.

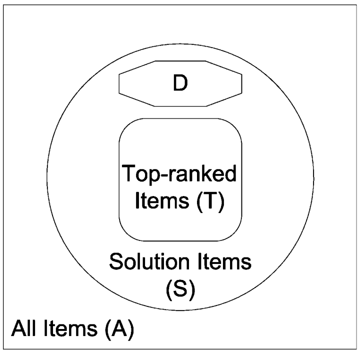


Fig. 8. Adding decoy elements.

solution items $D \subseteq S - T$ such that the occurring effects are neutralizing each other (see Fig. 8). Following this approach, finding a suitable set D can be defined as a minimization problem, i.e. *find D such that:*

$$\sum_{i \in P} |\mathrm{SIM}(i) - \mathrm{SDM}(i)| \rightarrow \mathrm{MIN}$$

with $P = T + D$ *or* $P = T - D,$ *respectively.*

3.3. Algorithmic design

The formulation of algorithms for decoy minimization is based on the approaches discussed in Section 3.2. Algorithm 1 implements the approach of removing items from a set of top-ranked items. The input consists of a list of top-ranked items and the minimal number of items to be presented (line 02). The difference between the size of the top-ranked items and the minimal size of the presented items constitutes the maximal size of the item set which can be removed (line 03). If the maximal size is equal or be-

```
01.FUNCTION decMinRemove RETURNS ItemList
02.PARAMETER: ItemList:topItems, Number:minPresented
03.  Number:maxSubSetSize <- sizeOf(topItems)-minPresented
04.  IF maxSubSetSize < 1 THEN
05.     RETURN topItems
06.  END IF
07.  Number:minDiff <- ∞
08.  ItemList:resultList <- {}
09.  FOR EACH Set:subSet <- getSubSet(topItems, maxSubSetSize)
DO
10.     ItemList:currentList <- topItems-subSet
11.     Number:currentDiff <- calcSimSdmDiff(currentList)
12.     IF currentDiff < minDiff THEN
13.       resultList <- currentList
14.       minDiff <- currentDiff
15.     END IF
16.  END FOR EACH
17.  RETURN resultList
18.END FUNCTION
```

Algorithm 1. Algorithmic design for decoy minimization by removing items.

low zero no items can be removed from the top-ranked items as they all are to be displayed. In this case the input set is also the output (lines 04–06). Otherwise, the algorithm iteratively creates all possible subsets of the top-ranked items by removing maximally *maxSubSetSize* items from the top-ranked items (lines 09–16). Thereby, the smallest difference between SIM and SDM and the corresponding set are calculated (lines 10 and 11) and stored (lines 07–08 and 12–15). The *resultList* is a purified (i.e. debiased) list of top-ranked items (line 17).

In Algorithm 1 also the items with the highest utilities are removed, if this leads to the minimum difference between SIM and SDM. This is not always desirable. In such cases, Algorithm 2 should be preferred. Algorithm 2 implements the approach of starting with a set of top-ranked items which will definitely be presented and extend this set by adding a limited number of further solution items. The input consists of a set of top-ranked items, a set of further solution items, and the maximal number of presented items (line 02). The difference between the maximal number of presented items and the size of the set of top-ranked items constitutes the maximal number of items to be added (line 03). Solution-items which are already in the set of top-ranked items cannot be added again and therefore are excluded from *solutionItems* (line04). After that, the algorithm iteratively creates all possible subsets of the extended top-ranked items by adding maximally *maxSubSetSize* items from the solution-items (lines 07–14). Thereby, the minimum difference between SIM and SDM and the cor-

```
01.FUNCTION decMinAdd RETURNS ItemList
02.PARAMETER: ItemList:topItems, ItemList:solutionItems,
Number:maxPresented
03.  Number:maxSubSetSize <- maxPresented-sizeOf(topItems)
04.  solutionItems <- solutionItems-topItems
05.  ItemList:resultList <- {}
06.  Number:minDiff <- ∞
07.  FOR EACH Set:subSet <- getSubSet(solutionItems,
maxSubSetSize) DO
08.    ItemList:currentList <- topItems+subSet
09.    Number:currentDiff <- calcSimSdmDiff(currentList)
10.    IF currentDiff < minDiff THEN
11.      resultList <- currentList
12.      minDiff <- currentDiff
13.    END IF
14.  END FOR EACH
15.  RETURN resultList
16.END FUNCTION
```

Algorithm 2. Algorithmic design for decoy minimization by adding items.

responding set are calculated (lines 08 and 09) and stored (lines 05–06 and 10–13). The *resultList* constitutes an extension of the set of top-ranked items (line 15).

Although the search for a set of items to be added or removed is bounded by the minimal respectively maximal number of presented items the building of of all possible decoy sets can be very expensive for large item databases. A possible way out of this problem is to accept near optimal solutions. Algorithm 3 shows the sketch of a hill-climbing version of Algorithm 2. First, all the top-ranked items are included in the set of result items (i.e. *resultList*, line 05). Subsequently, in each iteration (lines 07–26) the item which decreases the difference between SIM and SDM the most (lines 10–17) is added to *resultList* (lines 20–25). The algorithm stops iterating and extending the result set when either *maxSubSetSize* is reached (lines 07 and 24) or no remaining solution-item further decreases the difference between SIM and SDM of the result set (lines 18–20).

Each of these algorithms has its strengths and weaknesses. Algorithms 1 and 2 produce optimal solutions whereas Algorithm 3 does not. On the another hand Algorithm 3 is much cheaper in terms of computing costs. Comparing Algorithms 1 and 2, Algorithm 1 is cheaper but Algorithm 2 can guarantee that certain items (e.g. with the highest utilities) remain in the result set. Which of the presented approaches should be applied and how the concrete implementation in a system looks like is depending on many practical factors, for example:

```
01.FUNCTION decMinHill RETURNS ItemList
02.PARAMETER: ItemList:topItems, ItemList:solutionItems,
Number:maxPresented
03.  Number:maxSubSetSize <- maxPresented-sizeOf(topItems)
04.  solutionItems <- solutionItems-topItems
05.  ItemList:resultList <- topItems
06.  Number:minDiff <- calcSimSdmDiff(resultList)
07.  WHILE maxSubSetSize > 0 DO
08.    Item:currentItem <- NULL
09.    Number:currentDiff <- minDiff
10.    FOR EACH Item:item IN solutionItems DO
11.      ItemList:currentList <- resultList+item
12.      Number:diff <- calcSimSdmDiff(currentList)
13.      IF diff < currentDiff THEN
14.        currentItem <- item
15.        currentDiff <- diff
16.      END IF
17.    END FOR EACH
18.    IF currentItem = NULL THEN
19.      RETURN resultList
20.    ELSE
21.      solutionItems <- solutionItems-currentItem
22.      minDiff <- currentDiff
23.      resultList <- resultList + currentItem
24.    maxSubSetSize <- maxSubSetSize - 1
25.    END IF
26.  END WHILE
27.  RETURN resultList
28.END FUNCTION
```

Algorithm 3. Hill-climbing approach for decoy minimization.

- the recommendation domain,
- the concrete implementation of the recommender system,
- time limitations,
- size of item database,
- offline pre-calculation of result sets vs. online calculation in interactive settings.

4. User experiments

The asymmetric dominance effect (ADE) is among the decoy effects the most stable and best controllable effect and as such best qualified for examining the possibilities of decoy minimization in online decision making. To this end, we conducted an unsupervised online user study consisting of two different experiments, whereby the first one is two-dimensional and the second one is three-dimensional in terms of the number of attribute dimensions.

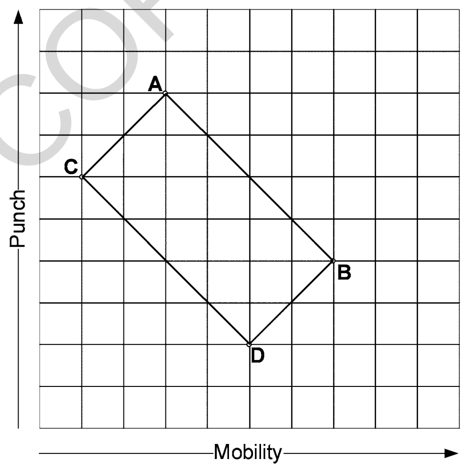
To additionally support the notion of competitive concurrency in which items typically are when they are subject to purchase we chose *action avatars in com-*

Fig. 9. Screenshot of the two-dimensional experiment.

puter games as domain for the selection task. Subjects were asked to estimate the result of a fight tournament. To this end, action avatars (items) with different strengths and weaknesses were presented (see Fig. 9). The avatars were described by the dimensions *punching power* and *mobility/quickness* on a scale 0 (very bad) – 10 (very good). In order to suppress influences of the avatar's picture representation the same picture was used for every option except that they were marked randomly with different colors. Subjects were randomly assigned to item sets (set size 2–4) consisting of different combinations of the items shown in Fig. 10. The items were designed such that there were two superior items of equal (SIM, MAUT) utility (A: $8 + 3 = 11$, B: $4 + 7 = 11$), when equal dimension weights are assumed (i.e. mobility and punch equally important). For both A and B a designated decoy item was designed which should trigger the ADE. Item C served as a decoy for A (i.e. C is totally dominated by A but not by B) and item D served as a decoy for item B.

The following hypotheses were formulated for the two-dimensional experiment:

- H1: Reproducing the decoy effect: Item A is forecasted to win the tournament (i.e. rank = 1) in the set ABC more often than in the set AB.
- H2: Reproducing the decoy effect: Item B is forecasted to win the tournament (i.e. rank = 1) in the set ABD more often than in the set AB.
- H3: Concurrent display of both decoys leads to a minimization of decoy effects.



Item	Punch	Mobility
A	8	3
B	4	7
C	6	1
D	2	5

Fig. 10. Items of the two-dimensional experiment.

Table 1 is summarizing the winner forecasts of the subjects for the relevant experimental sets[1]. Both decoys triggered an effect which resulted in an increase of the frequency of how often the dedicated target was predicted to win the tournament. Hence, H1 and H2 can be confirmed. In order to examine H3 we have

[1] The data set has been purified by omitting sessions where a decoy had been forecasted to be the winner. This did not change the results significantly, but it seems to adequate as there is no rational reason for such a choice.

Table 1

Results of the two-dimensional experiment

Set	Winner A	Winner B	Total
AB	23	43	66
	34.8%	65.2%	100%
ABC	30	36	66
	45.5%	54.5%	100%
ABD	14	47	61
	23.0%	77.0%	100%
ABCD	20	46	66
	30.3%	69.7%	100%
Total	87	172	259
	33.6%	66.4%	100%

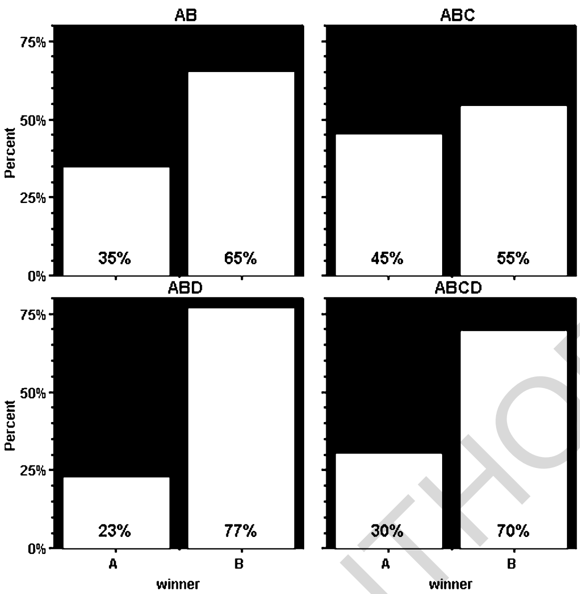
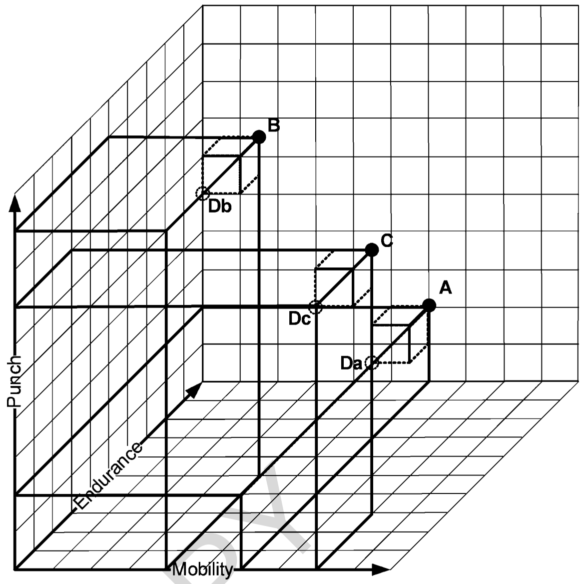


Fig. 11. Results of the two-dimensional experiment.

to compare all the sets AB, ABC, ABD, ABCD (see Fig. 11). It becomes obvious that the concurrent display of both decoys resulted in mutual neutralization such that the outcome of the set ABCD nearly matches the outcome of the set AB.

To generalize the findings of the two-dimensional experiment, a second experiment with different set sizes and higher attribute dimension cardinality was carried out. The domain (fight tournament and action avatars) was kept the same, but now the different items were described by *punching power*, *mobility/quickness*, and *endurance*. The subjects were told that the punching power and mobility is decreasing during a fight depending on the endurance of the avatar (i.e. the better the endurance the smaller is the decrease of punch and mobility). There were three superior items (A, B, C) of equal utility (SIM, MAUT), when equal dimension weights are assumed. Additionally, there was a corresponding decoy item for A (Da), B (Db), and C (Dc).



Item	Punch	Mobility	Endurance
A	2	6	10
B	9	4	5
C	7	8	3
Da	1	5	9
Db	8	3	4
Dc	6	7	2

Fig. 12. Items of the three-dimensional experiment.

Figure 12 is summarizing the experimental design and the corresponding item values. The task for the subjects was similar as in the two-dimensional experiment. Subjects had to state which of the presented avatars is going to win.

The following hypotheses were formulated for the three-dimensional experiment:

– H1: Reproduction of decoy effect: More people forecast item A as the winner when Da is additionally presented.
– H2: Reproduction of decoy effect: More people forecast item B as the winner when Db is additionally presented.
– H3: Reproduction of decoy effect: More people forecast item C as the winner when Dc is additionally presented.
– H4: Concurrent display of Da and Db lead to a minimization of the corresponding decoy effects.

– H5: Concurrent display of Da and Dc lead to a minimization of the corresponding decoy effects.
– H6: Concurrent display of Db and Dc lead to a minimization of the corresponding decoy effects.

Table 2 lists the results for all investigated experimental sets. As it is shown in Fig. 13, the decoy effects could not be reproduced generally as intended (H1–H3). For the set ABDbC compared to ABC the decoy effect could be reproduced (H2), but in the set ADaBC the item Da tendentially acted as a decoy for C rather than for A. As well in the set ABCdc the decoy slightly acted as a decoy for B rather than for C. As in a complex setting always more than one effects are occurring, in these two cases other decoy effects like AE (i.e. Da for C) and CE (i.e. Dc for B) obviously were felt stronger by the subjects than the intended ADEs or at least had a big unintended influence.

However, when comparing the simpler decoy groups of size three and the corresponding decoy minimization group, some impressing results reinforcing the findings of the two-dimensional experiment are revealed. As Fig. 14 shows, for all three groups of sets (i.e. two different decoys plus concurrent display of both decoys) the concurrent display of the decoys (i.e. decoy minimization) resulted in mutual compensation of the effects. Figure 14 reveals that all choice distributions of the decoy minimization groups lie between the distributions of the corresponding decoy groups. Thus, decoy minimization could be reproduced in all cases and H4–H6 can be supported.

Table 2
Results of the three-dimensional experiment

Set	Winner A	Winner B	Winner C	Total
ABC	39	69	15	123
	31.7%	56.1%	12.2%	100%
ADaBC	20	36	11	67
	29.9%	53.7%	16.4%	100%
ABDbC	9	27	5	41
	22.0%	65.9%	12.2%	100%
ABCDc	11	30	3	44
	25%	68.2%	6.8%	100%
ADaB	33	26		59
	55.9%	44.1%		100%
ABDb	12	33		45
	26.7%	73.3%		100%
ADaBDb	20	25		45
	44.4%	55.6%		100%
BDbC		27	15	42
		64.3%	35.7%	100%
BCDc		21	21	42
		50.0%	50.0%	100%
BDbCDc		27	23	50
		54.0%	46.0%	100%
ADaC	33		18	51
	64.7%		35.3%	100%
ACDc	18		30	48
	37.5%		62.5%	100%
ADaCDc	30		25	55
	54.5%		45.5%	100%
Total	225	321	166	712
	31.6%	45.1%	23.3%	100%

5. Influence of decoy minimization on uncertainty levels

Decoy effects, especially the asymmetric dominance effect, can have a significant impact on the perceived uncertainty during a decision task. In [20] it is shown that decoy effects increase decision confidence (i.e. decrease uncertainty). In order to investigate how the concept of decoy minimization interacts with the influence of the decoys we also recorded the self reported uncertainty levels after every decision task in the experiments discussed in the last section. The main question was whether there was a way of decreasing the uncertainty during decision making by the introduction of decoys and concurrently maintaining a maximum of objectivity by the addition of counteracting decoys (i.e. decoy minimization).

Table 3 summarizes the mean uncertainty in the different groups of the two-dimensional experiment.

Table 3
Uncertainty levels of the experimental groups in the two-dimensional experiment

Set	Mean uncertainty	N	Std. Deviation
AB	3.41	66	1.509
ABC	3.06	66	1.762
ABD	3.03	61	1.303
ABCD	3.56	66	1.560

Confirming previous results the uncertainty decreases when a decoy is presented (i.e. compare AB to ABC and ABD). This effect is reversed when a counter acting decoy (i.e. decoy minimization in ABCD) is presented. This result is also supported by the uncertainty levels in the three-dimensional experiment. Table 4 is showing the mean uncertainty for all decoy and corre-

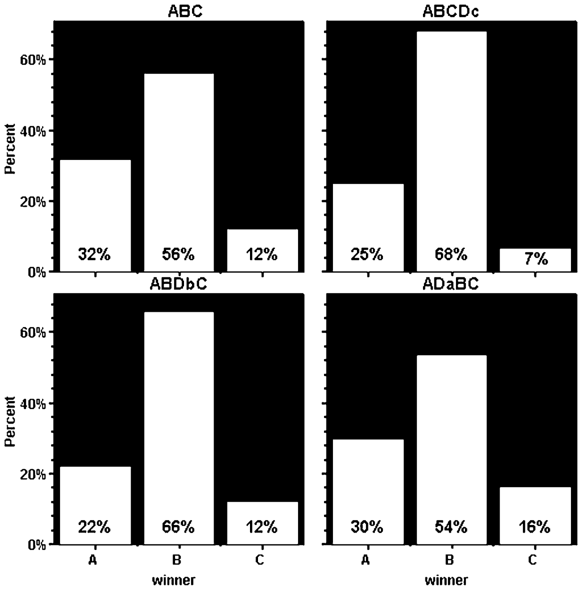


Fig. 13. Results of the three-dimensional experiment: reproduction of the decoy effect (H1–H3).

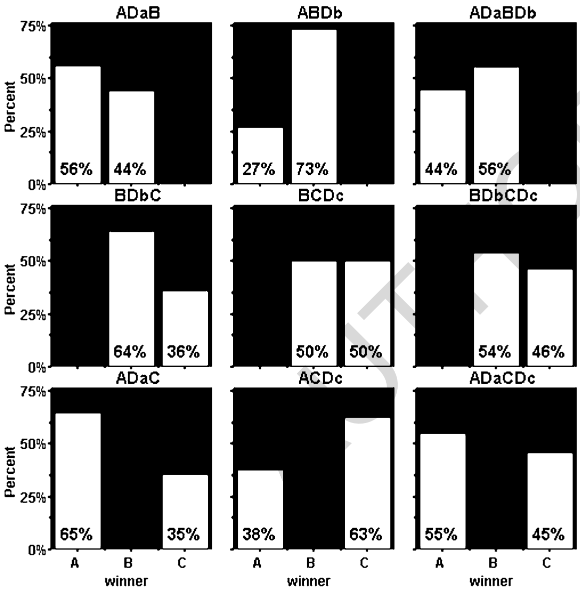


Fig. 14. Results of the three-dimensional experiment: decoy minimization (H4–H6).

sponding decoy minimization groups. In all cases the decoy minimization condition triggers higher average uncertainty than the corresponding decoy conditions.

The most important conclusion to draw at this point is that the decrease of the uncertainty levels because of a decoy does not remain when additional counteracting decoys are added. This means that decoy effects

Table 4

Uncertainty levels of decoy and decoy minimized groups in the three-dimensional experiment

Set	Mean uncertainty	N	Std. Deviation
ADaB	3.49	59	1.633
ABDb	3.22	45	1.347
ADaBDb	3.67	45	1.665
BDbC	3.21	42	1.881
BCDc	3.12	42	1.468
BDbCDc	3.46	50	1.515
ADaC	3.08	51	1.573
ACDc	3.44	48	1.500
ADaCDc	3.69	55	1.489

exploited to favor one item can result in alleviated decision making but the restoring of objectivity by decoy minimization again increases the decision dilemma.

6. Conclusions and future work

Decoy effects are omni-present on recommender result pages but state-of-the-art systems are completely unaware of such effects. As such cognitive effects trigger irrational and therefore potentially suboptimal decision making, it is necessary to equip decision support systems in general and recommender systems in particular with mechanisms which allow to reduce misleading decoy effects. This paper introduced the concept of decoy minimization which constitutes a utility-based recommender approach for detecting and neutralizing decoy effects on recommender result pages. Two user experiments have been reported which clearly show the impact of decoys on rational decision making and come up with a first proof of concept for the decoy minimization approach.

Furthermore, it could be shown that, although decoy effects decrease the uncertainty levels during decision making, decoy minimization increases the uncertainty levels again. In other words, it is not possible to decrease uncertainty and restore a maximum of objectivity by means of decoys at the same time.

Future work concentrates on one hand on the generalization of the decoy minimization approach on other types of decoy effects and on the other hand on the development of statistical methods for decoy model learning. The idea is to use recommender user logs (i.e. recorded choice sets and answers) in order to estimate domain specific and therefore more accurate decoy models automatically. This would build the basis of implementing a decoy filter functionality for recommender- and decision support systems which serves the purpose of increasing decision quality.

References

[1] D. Ariely and T. Wallsten, Seeking subjective dominance in multidimensional space: An exploration of the asymmetric dominance effect, *Organizational Behaviour and Human Decision Processes* **63**(3) (1995), 223–232.

[2] R. Burke, Hybrid recommender systems: Survey and experiments, *User Modeling and User-Adapted Interaction* **12**(4) (2002), 331–370.

[3] J.R. Busemeyer and J.T. Townsend, Decision field theory: A dynamic cognitive approach of decision making in an uncertain environment, *Psychological Review* (100)(1992), 432–459.

[4] M. Devetag, From utilities to mental models: A critical survey on decision rules and cognition in consumer choice, *Industrial and Corporate Change* **8**(2) (1999), 289–351.

[5] A. Díaz and P. Gervás, Personalisation in news delivery systems: Item summarization and multi-tier item selection using relevance feedback, *Web Intelligence and Agent Systems* **3**(3) (2005), 135–154.

[6] A. Felfernig, G. Friedrich, D. Jannach and M. Zanker, An environment for the development of knowledge-based recommender applications, *International Journal of Electronic Commerce (IJEC)*, **11**(2) (2006), 11–34.

[7] A. Felfernig, S. Gordea, D. Jannach, E. Teppan and M. Zanker, A short survey of recommendation technologies in travel and tourism, *ÖGAI Journal* **4**(25) (2006), 17–22.

[8] A. Felfernig, B. Gula and E. Teppan, Knowledge-based recommender technologies for marketing and sales, *International Journal of Pattern Recognition and Artificial Intelligence* **21** (2007), 333–355.

[9] G. Häubl and K.B. Murray, Processes of preference construction in agent-assisted online shopping, in: *Online Consumer Psychology: Understanding and Influencing Behavior in the Virtual World*, ISBN-10: 0805851550, Lawrence Erlbaum Associates Inc., pp. 265–286, 2005.

[10] G. Häubl and V. Trifts, Consumer decision making in online shopping environments: The effects of interactive decision aids, in: *Online Consumer Psychology: Understanding and Influencing Behavior in the Virtual World Marketing Science*, Vol. 19, 2000, Special Issue on Marketing Science and the Internet, pp. 4–21.

[11] J.L. Herlocker, J.A. Konstan and J. Riedl, Explaining collaborative filtering recommendations, in: *Proc. of the ACM Conference on Computer Supported Cooperative Work*, Pennsylvania, USA, 2000, pp. 241–250.

[12] D. Kahneman, Maps of bounded rationality: Psychology for behavioral economics, *The American Economic Review* **93**(5) (2003), 1449–1475.

[13] M. Pazzani and D. Billsus, Learning and revising user profiles: The identification of interesting web sites, *Machine Learning* **27** (1997), 313–331.

[14] C. Schmitt, D. Dengler and M. Bauer, Multivariate preference models and decision making with the maut machine, in: *User Modeling 2003*, LNCS, Vol. 2702, Springer-Verlag, Berlin/Heidelberg, 2003, pp. 297–302.

[15] D.L. Schwarzkopf, The effects of attraction on investment decisions, *Journal of Behavioural Finance* **4**(2) (2003), 96–108.

[16] H.A. Simon, A behavioral model of rational choice, *The Quarterly Journal of Economics* **69**(1) (1955), 99–118.

[17] I. Simonson, Choice based on reasons: The case of attraction and compromise effects, *Journal of Consumer Research* **16**(2) (1989), 158–174.

[18] I. Simonson and A. Tversky, Choice in context: Tradeoff contrast and extremeness aversion, *Journal of Marketing Research* **39** (1992), 281–292.

[19] I. Simonson and A. Tversky, Context-dependent preferences, *Management Science* **39**(10) (1993), 1179–1189.

[20] E. Teppan and A. Felfernig, The asymmetric dominance effect and its role in e-tourism recommender applications, in: *Proc. of the 9th Wirtschaftsinformatik Conference*, Vienna, Austria, 2009, pp. 791–800.

[21] E. Teppan and A. Felfernig, Calculating decoy items in utility-based recommendation, in: *Proc. of the 22nd International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems*, Tainan, Taiwan, 2009, pp. 183–192.

[22] E. Teppan and A. Felfernig, Asymmetric dominance- and compromise effects in the financial services domain, in: *Proc. of the 12th IEEE Conference on Commerce and Enterprise Computing*, Shanghai, China, 2009, pp. 57–64.

[23] E. Teppan and A. Felfernig, Impacts of decoy elements on result set evaluations in knowledge-based recommendation, *International Journal of Advanced Intelligence Paradigms (IJAIP)* **1**(3) (2009), 358–373.

[24] M. Vozalis and K.G. Margaritis, On the enhancement of collaborative filtering by demographic data, *Web Intelligence and Agent Systems*, **4**(2) (2006), 117–138.

[25] D. Winterfeldt and W. Edwards, *Decision Analysis and Behavioral Research*, ISBN-10: 052125308X, Cambridge University Press, Cambridge, New York, 1986.