Efficient Preference Elicitation with Coarse Preferences

IJCAI 2017 Submission #3703

Abstract

We present a new model for the representation of preferences that are based on coarse criteria, leading to better recommendations in problems where user behaviour is consistent with them making decisions by aggregating many individual options. When this is the case, our '*coarse preference*' model enables an elicitation procedure that outperforms the state-of-the-art utility elicitation procedure that is based on the assumption of additive independence of variables. We demonstrate this through experiments performed with a real-world data set derived from a mobile clothes shopping app. We show that our proposed model achieves double the recommendation acceptance rate and a significant increase in platform profit, due to the use of a decomposition that is better suited to the true nature of the users' indifference regions. Furthermore, our model is of lower complexity than similar additive independence models, with computation time rising at a slower rate with respect to dimensionality of item descriptions - achieving a halving of computation time in our experiment. Our results have insights for recommendation systems designers, potentially enabling new ways of representing user utility when secondary criteria need to be maximised without compromising performance.

1 Introduction

Many everyday economic transactions are gradually shifting to the digital sphere. The availability of vast product catalogs, along with the opportunity of increased revenue through automation, has led to the use of *recommender systems* [Ricci *et al.*, 2010] in many application domains, especially those involving large user populations on the Web. These systems aim at providing users with relevant options or suggest products that are likely to lead to a purchase. In order to do so, they attempt to learn models of how users react to the recommendation of a product.

As products are defined in terms of sets of features and their values, the models used to represent users' preferences often involve "factored" representations [Braziunas, 2006; Ricci *et al.*, 2010], e.g. that the value a user ascribes to a camera is a linear combination of partial utilities corresponding to the value the user places on the camera's price, its resolution, and zoom factor. Given the number of items and users featured on online service providers, any concrete user-item preference model built from realistic amounts of observations will be inevitably sparse, so it is only natural that learning models will aim to associate properties of items with users' preferences in order to generalise over individual options.

Evidence coming from Cognitive Science, on the other hand, suggests that users often only distinguish between rather *coarse-grained* categories of choices, and would not necessarily evaluate choices by scrutinising every individual instantiation down to their descriptive features [Rosch and Lloyd, 1978; Wilson and Keil, 2001]. For example, the user above might be indifferent towards the price of the camera, as long as it falls within their budget and has 'sufficient' zoom factor. We expect that models that are able to identify these groups of products will be better equipped to accurately predict user responses to recommendations. The more concise model can reduce the number of interactions needed to learn a good representation of the users' preferences, while also allowing the system to exploit their relative indifference between choices. The latter could be useful for applications that exploit forms of 'nudging' [Leonard, 2008], or those where user satisfaction is crucial but the system needs to consider other criteria in order to guarantee its own functionality. This is the case, for example, in sharing economy applications like AirBnB, where services can only be provided if sufficient commission earnings are generated, which, in turn, depend on enough rental deals being closed.

We develop a novel model of coarse user preferences, and accompanying preference elicitation procedure, for problems where the users' behaviour in a domain is consistent with them grouping their options into sets of equally preferred items. Crucially, we do not assume access to methods for explicitly asking the user about these categories, nor that the user has explicit awareness of them. We show that our coarse preference model can outperform state-of-the art utility-based preference elicitation algorithms when applied to online sequential recommendation applications. This improvement materialises in terms of user and system utility generated, as well as in terms of computation time and learning speed. We demonstrate this by presenting experimental results on a real-world data set acquired from a commercial mobile clothes shopping application called Mallzee [1] [Guan *et al.*, 2016; Tattan, 2016].

2 Related Work

In order to achieve compact representations of preferences, utility functions are usually defined over a multivariate space of item descriptors [Braziunas, 2006]. If preferences over this space exhibit suitable structure, a preference relation can be modeled more concisely, and utility functions can be decomposed into sub-utility functions [Braziunas, 2006]. The typical approach to utility function decomposition is to assume that it exhibits some form of preferential independence, usually that of *Additive Independence* (AI) [Keeney and Raiffa, 1993] or *Generalised Additive Independence* [Gonzales and Perny, 2004]. Although these models successfully capture the effect different variable assignments have on the user's preferences, they fail to capture dependencies within their, individual or combined, variable domains. [2]

Boutilier et al. [2009] presented an online feature elicitation procedure where, given a known utility function over *concepts*, essentially subsets of the outcome space that satisfy some specific formulas, uncertainty over the specification of concepts was reduced sufficiently to enable an optimal decision. They expand on this in [Boutilier *et al.*, 2010] by introducing uncertainty over the utility function. Though we are interested in eliciting utility functions over a coarser representation of the outcome space, which a set of concepts could be, we do not require the user to label these subspaces. Godoy-Lorite et al. [2016] presented a non-utility based model which takes advantage of groupings of items, but do so in a way which can not capture the user's indifference between alternatives, since items can belong to multiple groups. Moreover, their model does not generalise to new, unrated, items.

3 Problem Description

Our envisioned system is tasked with sequentially presenting a user with product recommendations of a set of items, e.g. clothes, while receiving feedback from the user after each such recommendation in the form of a rating of the presented item. Our aim is to learn what items the user prefers in as few interactions as possible, with satisfactory accuracy, such that future recommendations are optimised with respect to our objective function. We therefore approach our problem as one of *Preference Elicitation*.

The system represents items in a multivariate vector space X. Variables in this space could, e.g., refer to the type of clothing, its brand, price, or colour. At each time-step t the user is presented with an item $x_t \in Q_t \subseteq X$, where Q_t is a set of available items for recommendation at time t. Each recommendation x_t elicits a response $r_t \in R$, where R is the space of possible responses. The user evaluates an item $x \in X$, according to a utility function $u : X \to \mathbb{R}$, with $u \in U$ drawn from the space of possible utility functions U. Values $u(x)$, $\forall x \in X$, define how much the user prefers each item. In other words, $u(x) > u(x') \Leftrightarrow x \succ x'$, $\forall x, x' \in X$ [von Neumann and Morgenstern, 1953]. We assume that the user's responses are truthful but noisy evaluations of the presented items, drawn from a Normal distribution $P_r = P(r_t|x_t, u) = \mathcal{N}(u(x_t), \sigma)$.

We consider two alternative *optimality criteria* for the system: maximising the *user's utility* over a recommended item, and maximising the *revenue* of the system from that recommendation. We are particularly interested in the trade-off between these two criteria, since they represent a conflict of interest between user and system. We define revenue as a constant percentage α of a recommended item's value multiplied by its rating by the user: $v(x, r) = \alpha \cdot r \cdot p(x)$, where $p(x)$ is the price of item x. The assumption is that accepted recommendations have a constant chance of translating into sales, and that the system receives a set commission on each sale. Maximising the user's utility is equivalent to maximising the user's rating of the recommended item, which is the focus of the majority of recommender system applications. Additional criteria are also often considered, such as *serendipity* and *diversity* [Andreadis *et al.*, 2016]. While we could incorporate these criteria into the user's utility function, this goes beyond the scope of this paper.

3.1 Preference Elicitation

When our aim is to maximise the user's utility, the optimal decision w.r.t. u is $x^* = \text{argmax}_{x\in X}\ u(x)$, giving utility $u(x^*)$. When optimising for revenue, the optimal decision w.r.t. u is $x^* = \text{argmax}_{x\in X} \int v(x, r)\ P_r dr$, with expected revenue $\int v(x^*, r)\ P_r dr$. This section assumes the former so as to simplify presentation.

Given a finite, and *manageable*[3], set of available items, we could assign one parameter for the utility of each item. When this is not the case, then more structured representations of the space U are used, such as those making use of additive [Koller and Friedman, 2009] or generalised additive independence [Gonzales and Perny, 2004] of variables. The system does not, in general, have complete knowledge of u, but maintains a density b^t over the space U, indicating its current belief over the user's utility function at time step t, with b^0 representing its prior knowledge. If we denote the expected utility of an outcome x given density b^t over U as $EU(x, b^t)$ then the optimal decision is $x^* = \text{argmax}_{x\in X} EU(x, b^t)$. We denote by $MEU(b^t)$ the value of being in state b^t, assuming one is forced to make a decision: $MEU(b^t) = EU(x^*, b^t)$.

At each time step t the system can present the user with an item $x_t \in Q_t$, eliciting a response $r_t \in R$. The user's response can be used to update our belief over their utility function, in accordance with Bayes' rule. The (myopic) *expected value of information (EVOI)* of a query can be defined by considering the difference between $MEU(b^t)$ and the expectation (w.r.t. r_t) of $MEU(b^{t+1})$. A myopically optimal elicitation strategy involves asking queries with maximal EVOI at each

[1] http://mallzee.com/

[2] In our experiments, we compare our approach to a variation of Guo et al. [2010], which we have adapted to our problem. This represents the state of the art in AI parametric utility models when users are not described by a set of parameters.

[3] We use *manageable* to refer to both memory-related performance and to the dynamics of the item population. This approach does not allow for generalising across items.

time step [Braziunas, 2006]. By presenting the user with the query that maximises the EVOI, we guarantee, in expectation, that a recommendation immediately after the user's response would provide them with the maximum possible utility.

4 Coarse Preferences

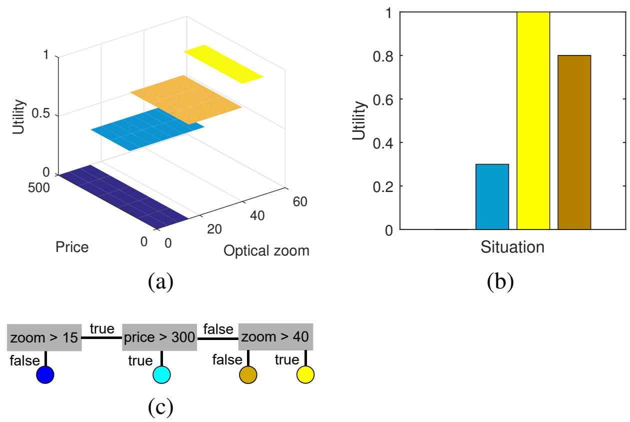

Figure 1: Representation of a user's coarse mapping for the camera example. (a) depicts the user's utility over different combinations of camera price and optical zoom factor. (b) depicts the equivalent utility function over the space of situations as defined by the partitioning represented by the decision tree in (c).

Key to our model of *coarse preferences* is the idea that a user's preferences are *coarser* than what the item vector space might imply. Specifically, it is often the case that his/her evaluation over an item is not as fine-grained as the combination of variable domains describing it but, rather, that they perceive alternatives in terms of the *situations* they represent. Intuitively, a user identifies in each item a corresponding situation, according to a mapping ϕ, and maintains a preference ranking over these situations rather than directly over the space of items. E.g. consider the user aiming to buy a new camera with their preferences being represented by the utility function in Figure 1a. We can partition the combined space of the variables "price" and "resolution" according to the decision tree in Figure 1c and represent the user's utility function in the space of situations in Figure 1b.

An alternative view of situations is that of *preferential equivalence classes* over the space of items. Essentially, every situation defines a subset of the space of items, all members of which are equally preferred. Intuitively, a user identifies in each item a corresponding situation, according to a mapping ϕ, and maintains a preference ranking over these situations rather than directly over the space of items. We define *coarse preferences* as follows:

Definition 4.1 (Coarse preferences)**.** We say that a decision maker exhibits *coarse preferences* ϕ over items $x \in X$ (or is a ϕ-coarse decision maker) if given a many-to-one mapping $\phi : X \rightarrow C$ from a space of items X to a space of *situations* C, with $\phi(x) = c, \phi(x') = c'$, we have:

$$x \succeq x' \Leftrightarrow c \succeq c' \tag{1}$$

Given a utility function $u : X \rightarrow \mathbb{R}$ defining a coarse mapping $\phi : X \rightarrow C$, we can therefore write the utility function

$$u^c : C \rightarrow \mathbb{R}, \tag{2}$$

with $u^c\left(\phi(x)\right) = u(x), \forall x \in X$.

4.1 Eliciting Coarse Preferences

If correct, the mapping ϕ from the space of items to that of situations allows us to select queries from the space of situations without loss of information. This reduces the cardinality of the query space from $|X|$ to $|C|$ and should speed up the convergence of the inference algorithm significantly compared to approaches defined over the original outcome space.

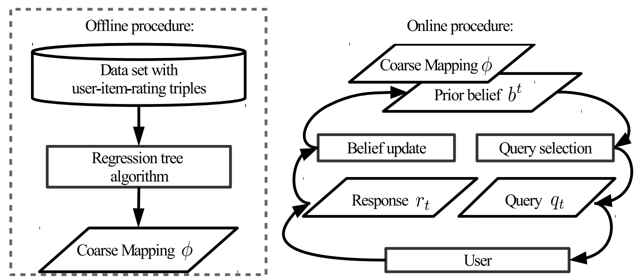

Figure 2: Overall methodology for the elicitation of coarse preferences. The offline procedure involves learning the coarse mapping ϕ from a corpus of data. The online preference elicitation procedure follows the normal procedure of query selection and belief updating to the user's response, but does so over the space of situations, as defined by ϕ.

We encode our belief over the user's utility for a situation c as a uni-variate Gaussian $\mathcal{N}\left(u^c(c); \mu(c), \sigma^2(c)\right)$. Since we are working on a discrete domain C, we can represent our belief over the utility function as

$$b^t = \prod_{c=1}^{|C|} \mathcal{N}\left(u^c(c); \mu(c), \sigma^2(c)\right). \tag{3}$$

Given a response r_t to a query $q_t = x_t$ at time step t, we can compute the new belief b^{t+1} according to Bayes' Rule. We write the posterior in closed form after each response as follows: $b^{t+1} \propto P_r \cdot b^t$.

So far, we have discussed how to elicit coarse preferences given a coarse mapping ϕ. We next present an offline methodology for learning this mapping from a corpus of user interactions, in the form of a set of ratings of products from past users of the system.

4.2 Learning the Coarse Mapping

In order to utilise the procedure detailed in the previous section, we need to first identify the coarse categories across our population of users. Though every user i will have their own mapping $\phi_i : X \to C_i$, the mapping ϕ resulting from their intersection can be used to make decisions for every individual user with no loss of utility since, as follows from Definition 4.1, a user i will exhibit both coarse preferences ϕ and ϕ_i.

We assume access to a history of user interactions in the form of user-outcome-rating triples, and that this history is representative of future users. We need to partition the solution space such that all outcomes in each class can be mapped to the same utility with little or no loss in accuracy. A clustering procedure which takes a target regression variable into account is Regression Tree Learning [Breiman *et al.*, 1984]. This algorithm receives a set of outcome-rating pairs as input, and outputs a decision tree with a continuous target variable; in this case, the utility. This decision tree defines a partitioning of the original outcome space by use of axis-parallel linear constraints, with each partition corresponding to a leaf node in the tree. It follows, that each of the tree's leaf nodes corresponds to a preferential equivalence class in the original outcome space X. Therefore, the decision tree can be used as a coarse mapping $\phi : X \to C$, with leaf nodes $c \in C$ representing the situations.

5 Experiments

We now describe our experimental procedure for evaluating our proposed model of *coarse preferences*, including a comparison against a state-or-the-art utility-based model by Guo et al. [2010].

5.1 Data set

We evaluate on a real-world data set generated from the interactions of users on Mallzee, a smart-phone clothing retail application. Users are identified solely by their *id*. Interactions are represented as binary responses (swiping left or right) to product recommendations. For each recommended item, we are provided with assignments to a set of 8 parameters (each variable's domain cardinality is given in parentheses), specifically: current price, discount from original price, currency (4), type of clothing (22), intended gender (5), whether it is in stock (2), brand (139), and colour (16). Out of these, *current price* and *discount from original price* are continuous, while all others are categorical variables. The data set consists of 200 users in total, each one responding to a different set of 500 products. We split the data into a *training* and *test* set, each comprising of the responses of 100 users. [4]

5.2 Procedure

We set uninformative priors for all methods, with an expectation of 0.5 utility spread equally across variables. The experiments are executed in two phases. First we learn the coarse mapping ϕ from our history of user interactions as stored in the *training* set. Then we run a series of 100 experiments with the *test* set acting as the query and recommendation space. We run one preference elicitation experiment instance for each of the users in the test set, for each algorithm and optimality criterion: expected user utility and system profit. For all methods, and at each time step t, we uniformly select 20 from the available queries for evaluation to comprise Q_t, representing real-world constraints, and motivated by computational reasons. We compute the EVOI for each query and select the maximising one to present to the user for evaluation. After receiving the user's response (as stored in the data set) we update our belief over the user's preferences. We evaluate the quality of our current belief state by hypothesizing an optimal recommendation from our available set of items, and noting the user's response from the data set. Under the user utility criterion, and when there are multiple items with the same expected evaluation, we select the one that would generate the most profit.

We compare our coarse preference elicitation procedure to the state of the art of AI-decomposed utility function elicitation procedure, as adapted from Guo et al. [2010] to account for ratings instead of pair-wise comparisons. In order to do so, we have to discretise the continuous product descriptors: *current price* and *discount from original price*. In this work, the belief takes the form:

$$b^t = \prod_{d=1}^{D} \prod_{i=1}^{|x_d|} \mathcal{N}\left(u^d(d_i); \mu(d_i), \sigma^2(d_i)\right), \tag{4}$$

where D is the dimensionality of the item representations, $|x_d|$ is the size of the corresponding discrete assignment space, and d_i is an assignment to variable d.

In order to learn the coarse mapping from the space of item descriptors to that of situations, we first transform all categorical variables into sets of binary variables. In order to determine the hyperparameters for the regression tree algorithm we randomly select a history of user interactions and run the offline clustering procedure for different values for the *maximum tree height* and *minimum number of nodes per leaf-node* hyperparameters, in increments of 1 and 50 respectively. The Regression Tree algorithm outputs an expected utility value at each leaf node. We round each prediction to its nearest integer value in order to get an understanding of how well the Decision Tree predicts the *training* data. We then chose the configuration which resulted in maximum precision when considering the products to which the users responded positively. During the experiments we then run a *regression tree* procedure as explained in Section 4.2, using these learned hyperparameters. To put these results into context, we perform least-squares linear regression on the same training set but without discretising the continuous variables, and round it as described above. We expect that this will give indication of how well the benchmark will perform.

Each leaf-node in this tree represents a situation. In order to be able to generalise to the test data, we wanted to keep the cardinality of the situation space relatively small compared to the available product space, focusing more on achieving higher precision, rather than recall, when considering the products to which users' responded positively. This is because making a good recommendation relies on our ability to identify a preferred item, rather than locating as many good items as possible.

To evaluate our performance for the criterion of user utility we plot the average normalised loss in user utility. This measures, at each time step, the normalised distance of the value assigned by the user to their recommended product, from the

[4] The data set is available at `http://bit.ly/2lwJZUP`.

maximum evaluation that could have been achieved, given the available products for recommendation. We also examine our performance in terms of system profit. For this purpose, we plot the normalised loss in profit, computed considering the value of the most expensive item available as the best possible profit achievable. This metric is agnostic to whether the user had swiped right or not on that item in the data set.

5.3 Results

Optimising for the regression tree hyperparameters resulted in setting a maximum height of 10, and a minimum number of 500 training samples per node. This in turn resulted in a space of 23 to 45 situations across experiment runs, with a mean of 33.2 and a standard deviation of 5.35 situations. Considering the products to which the user responded positively, we achieved average precision of 0.97 and average recall of 0.70 on the training data. The least-squares linear regression procedure resulted in an average precision of 0.81 and average recall of 0.21. Figure 3 shows part of a learned decision tree, with two examples of situations. The first one shows users grouping all products not displayed in British pounds into a single category. The second one refers to all non female-specific shirts, with a defined colour, and that are displayed in British pounds. These descriptions are easy to understand and put into context, and can act as an additional tool for system designers.

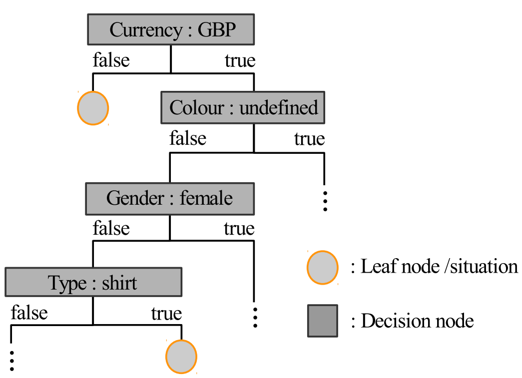

Figure 3: Part of a decision tree learned from the history of user interactions. Two leaf nodes are shown, each one identifying a situation in the space C.

Figure 4 plots the normalised loss of utility for the coarse preferences algorithm and the sum of conditional Gaussians benchmark, as adapted from [Guo and Sanner, 2010]. 'Time step' refers to the number of queries presented so far. The shaded areas cover ± 2 standard deviations. The lines labelled 'AI benchmark' present the results for the benchmark as run with each optimality criterion. Correspondingly, lines labelled "coarse preferences" present the results for our algorithm. The performance of our procedure is indicative of its ability to better trade-off between generality and accuracy for this data set. Querying any point in an equivalence class informs us of the user's preference for all its members. To the extent that this mapping is accurate, we achieve a significant reduction in the dimensionality of the solution space,

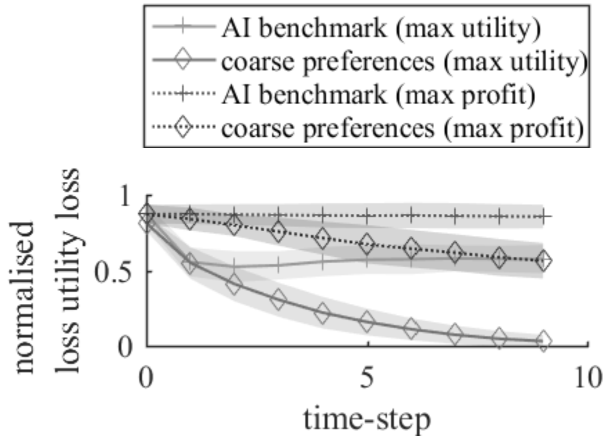

Figure 4: Average normalised loss of utility across 100 experiment instances from the user's response to our recommendation, averaged across 100 experiment runs.

by going from a space of 8 parameters with 500 samples (the space is combinatorially large but constrained by the number of available samples) to a space of 1 variable with a maximum size of $|C|$. This in turn translates to faster convergence to an accurate representation of the user's preferences. As expected, procedures optimising for user utility achieve better performance in this metric. However, even when optimising for user utility, the AI benchmark quickly converges to a suboptimal representation of user preferences. Crucially, when optimising for system profit, the users experience no improvement in their recommendations as time progresses. This results from the additive independent assumption being a bad fit to the specific problem combined with restrictions in its ability to query the user. These restrictions stem from the fact that not all of the solution space is available for querying with or, since we are using the same space, recommending to the user. In fact only 500 points out of a combinatorially large set are available. This problem is enhanced by further restricting the space to a randomly selected set of 20 available items. The coarse preferences approach is much more effective at generalising from a small number of samples and therefore does not suffer from this effect.

Figure 5 displays the gradual improvement in the effective monetary value of recommendations, in terms of average normalised loss. The gains in profits in comparison to the benchmark are significant, which we attribute to both our model being a better fit to the problem and our ability to select from a space of, from the user's perspective, equivalent solutions. One might expect that this graph would give a complementary image to that of Figure 4. However, both coarse preferences approaches significantly outperformed the AI benchmark. Moreover, running our procedure for optimal user utility outperformed the same procedure over system profit. We believe this to be the result of an overoptimistic prior over the utility of each situation, leading the algorithm to take risks for profit that don't always pay off. Focusing on the benchmark, we note that the improvement for the first two time-

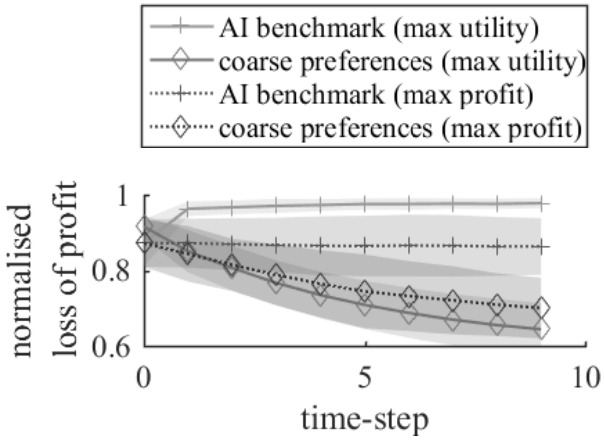

Figure 5: Average normalised loss of profit across 100 experiment instances from the user's response to our recommendation, averaged across 100 experiment runs.

steps when run for user utility does not translate to increased profits. Further, when optimising for profit the inability to accurately represent users' preferences translates to poor improvement in profit.

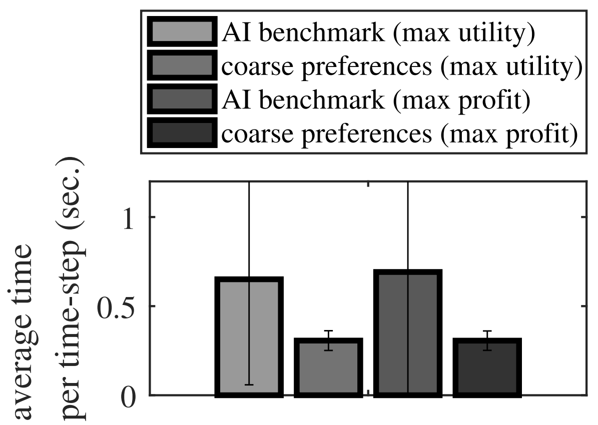

Figure 6: Average computational time in seconds for each preference elicitation model and optimality criterion.

Figure 6 presents the time taken during a single step of preference elicitation, averaged across all runs, experiments, and time steps. Error bars represent ± 2 standard deviations. The presented values include time taken for query evaluation and selection, as well as the belief update after the user's response. Regardless of optimality criterion, our approach requires about half as much time as the benchmark, which can be a significant advantage in online applications.

6 Discussion

The main incentive for adopting our approach is that it allows for using a different decomposition of utility functions, conditioned on different assumptions from those of variable-based decomposition models. As such, it expands the set of problems that can be sufficiently modelled by such approaches. Preference elicitation with coarse preferences is much faster on-line while their structure allows for optimising for secondary criteria, such as profit or environmental impact, without having to sacrifice in user utility. However, these benefits are hard to justify if user behaviour does not conform to the assumption of coarseness, i.e. that there are significant subspaces of the solution space in which users are indifferent between different solutions. Before deploying any utility function decomposition the system designer needs to verify whether its underlying assumptions approximate real user behaviour. Lastly, in its current iteration our model is dependent on having access to a history of user interactions, with the assumption that those will be representative of future user behaviour. This could potentially be circumvented by learning the coarseness online.

There exist a number of approaches for handling the exploration-exploitation trade-off, e.g. bandit algorithms [Busa-Fekete and Hüllermeier, 2014] and partially observable Markov decision processes [Boutilier, 2002]. The issue of computing these policies is orthogonal to that of exploiting the structure of the utility function representation.

7 Conclusion and Future Work

We propose a new approach to utility function decomposition termed *coarse preferences* which models user behaviour that is consistent with them evaluating alternatives based on which category each one falls into. Our approach is orthogonal to that of variable-based decompositions such as Additive Independence in that it can be used in combination with them while also being based on different assumptions. We demonstrate that there exist real-world problems where the coarseness assumption is a better fit than Additive Independence to user behaviour. This allows for a significant increase in recommendation quality while also taking advantage of reduced computational time; a benefit that scales with the number of variables. The magnitude of the effect our procedure had suggests that it is worthy of consideration in recommender system applications, particularly those that involve users rating sequentially presented items. Furthermore, our model is the first, to our knowledge, approach that allows for optimising for secondary criteria, such as profit, while guaranteeing to optimise for the user's utility.

Future extensions to this work could consider combining additive independence with coarse preferences, mixture models over coarse partitionings, the use of Gaussian process parametrisations [Davies and Ghahramani, 2014] and applications to practical domains such as ridesharing [Andreadis *et al.*, 2016].

Acknowledgements

"Space reserved for acknowledgements."

References

[Andreadis *et al.*, 2016] Pavlos Andreadis, Sofia Ceppi, Michael Rovatsos, and Subramanian Ramamoorthy. Diversity-aware recommendation for human collectives. In *European Conference on Artificial Intelligence Workshop on Diversity-aware Artificial Intelligence (DIVERSITY @ ECAI 2016)*, pages 23–32, the Hague, Netherlands, August 2016.

[Boutilier *et al.*, 2009] C. Boutilier, K. Regan, and P. Viappiani. Online feature elicitation in interactive optimization. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML-09, pages 73–80, Montreal, Quebec, Canada, 2009. ACM.

[Boutilier *et al.*, 2010] C. Boutilier, K. Regan, and P. Viappiani. Simultaneous elicitation of preference features and utility. In *Proceedings of the 24th National Conference on Artificial Intelligence*, AAAI-10, pages 1160–1197. AAAI, 2010.

[Boutilier, 2002] C. Boutilier. A POMDP formulation of preference elicitation problems. In *Proceedings of the 18th National Conference on Artificial Intelligence*, pages 239–246. AAAI, 2002.

[Braziunas, 2006] D. Braziunas. Computational approaches to preference elicitation. Technical report, University of Toronto, 2006.

[Breiman *et al.*, 1984] Leo Breiman, Jerome Friedman, Charles J Stone, and Richard A Olshen. *Classification and regression trees*. CRC press, 1984.

[Busa-Fekete and Hüllermeier, 2014] Róbert Busa-Fekete and Eyke Hüllermeier. A survey of preference-based online learning with bandit algorithms. In *International Conference on Algorithmic Learning Theory*, pages 18–39. Springer, 2014.

[Davies and Ghahramani, 2014] Alex Davies and Zoubin Ghahramani. The random forest kernel and other kernels for big data from random partitions. *arXiv preprint arXiv:1402.4293*, 2014.

[Godoy-Lorite *et al.*, 2016] Antonia Godoy-Lorite, Roger Guimerà, Cristopher Moore, and Marta Sales-Pardo. Accurate and scalable social recommendation using mixed-membership stochastic block models. *CoRR*, abs/1604.01170, 2016.

[Gonzales and Perny, 2004] Christophe Gonzales and Patrice Perny. GAI networks for utility elicitation. *KR*, 4:224–234, 2004.

[Guan *et al.*, 2016] Congying Guan, Congying Guan, Shengfeng Qin, Shengfeng Qin, Wessie Ling, Wessie Ling, Guofu Ding, and Guofu Ding. Apparel recommendation system evolution: an empirical review. *International Journal of Clothing Science and Technology*, 28(6):854–879, 2016.

[Guo and Sanner, 2010] Shengbo Guo and Scott Sanner. Real-time multiattribute bayesian preference elicitation with pairwise comparison queries. In *International Conference on Artificial Intelligence and Statistics*, pages 289–296. AISTATS, 2010.

[Keeney and Raiffa, 1993] R. L. Keeney and H. Raiffa. *Decisions with Multiple Objectives: Preferences and Value Trade-Offs*. Cambridge University Press, 1993.

[Koller and Friedman, 2009] D. Koller and N. Friedman. *Probabilistic graphical models: principles and techniques*. MIT Press, 2009.

[Leonard, 2008] Thomas C Leonard. Richard h. thaler, cass r. sunstein, nudge: Improving decisions about health, wealth, and happiness. *Constitutional Political Economy*, 19(4):356–360, 2008.

[Ricci *et al.*, 2010] F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor. *Recommender Systems Handbook*. Springer-Verlag New York, Inc., New York, NY, USA, 1st edition, 2010.

[Rosch and Lloyd, 1978] Eleanor Rosch and Barbara B Lloyd. *Cognition and categorization*. Lawrence Erlbaum Associates, Hillsdale, New Jersey, 1978.

[Tattan, 2016] Killian Villanova Tattan. Retail product recommender engine improvements. Master's thesis, University of Edinburgh, United Kingdom, 2016.

[von Neumann and Morgenstern, 1953] J. von Neumann and O. Morgenstern. *Theory of Games and Economic Behavior*. Princeton University Press, Princeton, 1953.

[Wilson and Keil, 2001] R. Wilson and F. Keil. *The MIT Encyclopedia of the Cognitive Sciences*. The MIT Press, Cambridge, MA, 2001.