NeurIPS 2023 Workshop on Adaptive Experimental Design and Active Learning in the Real World

Model-Free Preference Elicitation

Carlos Martin[*] CGMARTIN@CS.CMU.EDU
Carnegie Mellon University

Craig Boutilier CBOUTILIER@GOOGLE.COM
Ofer Meshi MESHI@GOOGLE.COM
Google Research

Abstract

Elicitation of user preferences is an effective way to improve the quality of recommendations, especially when there is little or no user history. In this setting, a recommendation system interacts with the user by asking questions and recording the responses. Various criteria have been proposed for optimizing the sequence of queries to improve understanding of user preferences, and thereby the quality of downstream recommendations. A compelling approach is *expected value of information (EVOI)*, a Bayesian approach which computes the expected gain in user utility for possible queries. Previous work on EVOI has focused on probabilistic models of user preferences and responses to compute posterior utilities. By contrast, in this work, we explore model-free variants of EVOI which rely on function approximation to obviate the need for strong modeling assumptions. Specifically, we propose to learn a user response model and user utility model from existing data, which is often available in real-world systems, and to use these models in EVOI in place of the probabilistic models. We show promising empirical results on a preference elicitation task.

Keywords: preference elicitation, expected value of information

1. Introduction

Recommendation systems (RSs) are crucial in making massive amounts of online content accessible to users in domains such as e-commerce, news, movies, videos, and others (Abel et al., 2011; Hallinan and Striphas, 2016; Linden et al., 2003; Pal et al., 2020; Covington et al., 2016). They typically leverage past user interactions to learn about a user's preferences and improve their future recommendations. However, in many cases, information about such preferences is lacking. For example, new users do not have enough interactions in their history. Alternatively, privacy considerations may prevent recording past interactions altogether. This is known as the cold-start problem (Lam et al., 2008; Bobadilla et al., 2012). In addition, past interactions may not always represent the user's taste accurately, such as when they share their account with others or their preferences change. In such cases, it may be challenging to accurately infer the user's true preferences.

Rather than relying solely on behavioral data, preference elicitation (PE) can be used to increase user agency by allowing them to communicate their true preferences (Keeney and Raiffa, 1976; Salo and Hamalainen, 2001; Rashid et al., 2008). In this setting, the RS

[*] Work done during an internship at Google Research.

presents queries to the user. Various types of queries can be used for PE. For example, we can ask the user about individual items ("Do you like movie X?") or comparisons ("Do you prefer movie X to Y?"). In this work, we focus on attribute-based queries (e.g., "Do you like science fiction movies?").

The central challenge, then, is how to select queries. This is a sequential decision problem, since the value of a query (hence its choice) at any point may be influenced by *subsequent* queries and responses. Several approaches for query optimization have been proposed in the literature. For example, *maximum information gain* uses a distribution over user preferences and selects the query whose expected response maximizes some measure of information (Rokach and Kisilevich, 2012; Zhao et al., 2018; Canal et al., 2019), as do related entropy-based methods Abbas (2004). Bourdache et al. (2019) proposes an approach based on Bayesian logistic regression. Alternative approaches include polyhedral/volumetric methods Iyengar et al. (2001); Toubia et al. (2003); ellipsoidal algorithms Salo and Hamalainen (2001) methods; and minimax-regret-based techniques Boutilier et al. (2006); Braziunas and Boutilier (2010); Boutilier (2013).

In this work we focus on *expected value of information (EVOI)*. EVOI selects queries by considering the expected improvement of downstream recommendation quality for each candidate query (Howard and Matheson, 1984; Guo and Sanner, 2010; Viappiani and Boutilier, 2010; Vendrov et al., 2020). It is a Bayesian approach that requires maintaining a probabilistic user model and computing a posterior distribution for user utility. For example, a user can be modeled as a multivariate Gaussian distribution in embedding space. However, even for such simple distributions, the posterior is intractable and requires approximations. As an alternative, we propose a *model-free* approach that does not make strong assumptions about the user distribution. Specifically, we show that it is possible to directly train predictive models from observational data and use their predictions for all required EVOI computations.

2. Model-Based Preference Elicitation

Let $\mathcal{Q}$ be the a set of queries, $\mathcal{R}_q$ be a set of responses to $q \in \mathcal{Q}$, and $\mathcal{X}$ be a set of items. Let $\mathcal{U} \subseteq \mathbb{R}^d$ be a set of users, modeled as points in an embedding space. The *user response model* specifies the probability $P(r|q,u)$ of $u \in \mathcal{U}$ responding to $q \in \mathcal{Q}$ with $r \in \mathcal{R}_q$, while the *user utility model* $v(x,u)$ specifies the utility of item $x \in \mathcal{X}$ to $u \in \mathcal{U}$. Initially, the RS starts with a prior belief $P(u)$. At each step of the session, the RS issues a query $q \in \mathcal{Q}$ and the user provides a response $r \in \mathcal{R}_q$. After observing a *history* of query-response pairs $H = ((q_1, r_1), \dots, (q_t, r_t))$, the RS updates its posterior belief $P(u|H)$ using Bayes' rule. If the utility function is linear, that is, $v(x,u) = x \cdot u$, then the expected utility of recommending $x \in \mathcal{X}$ is $\mathbb{E}_{u|H} v(x,u) = \mathbb{E}_{u|H} x \cdot u = x \cdot \big(\mathbb{E}_{u|H} u\big) = x \cdot \bar{u}$ where $\bar{u} = \mathbb{E}_{u|H} u$. Since it is optimal to recommend the item with the highest expected utility, the expected utility given a history H is $EU(H) = \max_{x \in \mathcal{X}} x \cdot \bar{u}$. If the posterior is modeled as a multivariate Gaussian, $\bar{u}$ is easy to compute.

This approach is *model-based*, since it assumes a specific form of the value function and posterior distribution. However, linearity and Gaussianity assumptions might be too simplistic to capture user preferences in realistic settings. For example, the mean of the user distribution might not capture the user's preferences well when the user has multiple interests in the item space (multimodality). Moreover, even with these simplifications, the posterior is still intractable to compute. In Section 3, we propose to explore more flexible, model-free representations of user utility and user response.

Computing optimal query strategies can be cast as a full-blown sequential decision making problem (Holloway and White, 2003; Boutilier, 2002). A simpler approach is to select queries myopically according to some criterion. One criterion which has been used successfully in previous work is EVOI (Chajewska et al., 2000; Guo and Sanner, 2010; Viappiani and Boutilier, 2010; Vendrov et al., 2020). To define EVOI, we first define the posterior expected utility (PEU) of a query: $PEU(q|H) = \sum_{r \in \mathcal{R}_q} P(r|H, q) EU(H; (q, r))$, where $H; (q, r)$ denotes H with (q, r) appended. The expected value of information is $EVOI(q|H) = PEU(q|H) - EU(H)$. (The query that maximizes $PEU(q|H)$ also maximizes $EVOI(q|H)$, since $EU(H)$ does not depend on q.) EVOI measures the improvement in expected utility offered by the query compared to not asking any query. This serves not only as a means for ranking potential queries, but also as a useful stopping criterion for elicitation. Notice that the two components required for computing PEU above are $P(r|H, q)$ and $EU(H)$. We next propose an alternative approach for their computation.

3. Model-Free Preference Elicitation

Our approach is motivated by real-world deployment of PE policies, where an existing policy is deployed and can be used to collect training data for new policies. In this setting, we have access to a dataset of *episodes*. Each episode contains a history H and the utility of the subsequent recommendation from the RS, as measured by user satisfaction with the recommended content (either by running user surveys or by using proxy measures such as engagement). Using this data, we can fit a function approximator to *directly* predict both $P(r|H, q)$ and $EU(H)$, which can be used to compute the PEU. Compared to the Bayesian approach in Section 2, this model-free approach obviates the need to model the full posterior $P(u|H)$, replacing posterior computation via Bayesian inference with model prediction. Apart from avoiding posterior approximations and unnecessary assumptions about user utility, this formulation facilitates sequential query optimization.

Furthermore, instead of doing one-step lookahead with respect to user response, we can do multi-step lookahead using algorithms like *depth-limited search (DLS)* and *Monte Carlo tree search (MCTS)* (Coulom, 2007), which lets the search tree grow asymmetrically toward more promising paths. Recent variants of MCTS use value function approximation to guide the search, including AlphaGo (Silver et al., 2016a), AlphaGo Zero (Silver et al., 2017), AlphaZero (Silver et al., 2018), MuZero (Schrittwieser et al., 2020), Gumbel Muzero (Danihelka et al., 2022), and Stochastic MuZero (Antonoglou et al., 2022). Stochastic MuZero, in particular, extends MuZero to plan with a stochastic model. This is useful in our setting, since user responses are stochastic. We use the open-source JAX implementation (Babuschkin et al., 2020) of Stochastic MuZero in our experiments. We could use a fully model-free RL approach, with the set of queries as the action space and the history as the state; however, fitting a separate response model lets us exploit the known structure of our PE problem (specifically, the space of qurey-response pairs) for planning. Planning at execution time has been shown to improve performance in many settings (Tamar et al., 2016; Guez et al., 2018, 2019; Farquhar et al., 2018; Oh et al., 2017; Silver et al., 2016b).

4. Experimental Setting

To evaluate our approach, we apply it to the movie recommendation domain with simulated users based on the MovieLens 1M dataset (Harper and Konstan, 2016). The dataset consists of movie ratings by users as well as a set of genres for each movie.[1] In our PE setting, the system presents queries to the user in the form "Do you like genre X?", and the user responds with either "yes" (1) or "no" (0). After a sequence of $T = 10$ rounds of elicitation,[2] the RS recommends a single movie and receives a response from the user indicating their satisfaction with the recommended movie.

To simulate user responses and utilities, we first use collaborative filtering to embed users, movies, and genres in a joint embedding space. Matrix factorization (Salakhutdinov and Mnih, 2007) is a common approach for computing embeddings. We solve an optimization problem of the form $\min_{U,X,Q} w_1\mathcal{L}_R(U^\top X, R) + w_2\mathcal{L}_A(X^\top Q, A) + w_3\|U\|_2^2 + w_4\|X\|_2^2 + w_5\|Q\|_2^2$ where U, X, Q, R, A are matrices representing the user embeddings, item embeddings, query embeddings, user-item ratings, and item-genre attributes (or relevances), respectively. As a preprocessing step, we rescale ratings to the unit interval using min-max normalization. We use $\mathcal{L}_R(x, y) = (\sigma(x) - y)^2$ as a regression loss for ratings, where σ is the logistic sigmoid function (used to map its real-valued input to the unit interval). $\mathcal{L}_A$ is a classification loss for attributes, and uses sigmoid binary cross-entropy loss. The last three terms are L_2 regularization terms.

We solve for U, X, Q using gradient descent. As shown in Figure 2, we obtain low regression and classification error. After generating user, movie and genre embeddings, we generate synthetic episodes by drawing queries from a base PE policy and a user response model. For simplicity, we employ a random policy for data generation, where the query is selected uniformly at random. In practice, a deployed elicitation policy will have a lower coverage of the query space than a random policy, but we use the latter here for illustrative purposes. The response of a user u to a query q is sampled from a Bernoulli distribution with parameter $\sigma(u \cdot q)$. Given a recommended movie $x \in \mathcal{X}$, a user's utility is computed as $u \cdot x$.[3] We assume a Bayesian RS which uses a multivariate Gaussian distribution to represent users, with an MCMC-based approximation for posterior update $P(u|H)$. To recommend a movie, it computes $\bar{u}$ (see Section 2) and returns the best movie, $\operatorname{argmax}_{x\in\mathcal{X}} x \cdot \bar{u}$. Notice that our approach is general and uses the RS as a black-box; therefore, any RS which recommends items based on histories H can be incorporated. Finally, each episode consists of the history H together with the utility to the user of the final recommended item.

We emphasize that the user embeddings are *only* used to generate the synthetic dataset. The elicitation system does *not* have access to the user embeddings.

5. Architectures

The data generated above is used to train a model that takes as input a history and outputs both a utility prediction and a response prediction for each possible subsequent query. That is, the model has type $(\mathcal{Q} \times \mathcal{R})^* \to \mathbb{R} \times (\mathcal{Q} \to \triangle\mathcal{R})$, where $\triangle\mathcal{R}$ is the set of probability

[1] The genres are: action, adventure, animation, children's, comedy, crime, documentary, drama, fantasy, film-noir, horror, musical, mystery, romance, sci-fi, thriller, war, and western.

[2] We use a fixed T here, but extending to variable T is straightforward by considering session abandonment.

[3] We use a linear utility for simplicity, but, as noted above, our approach supports non-linear utilities.

distributions on $\mathcal{R}$. (For binary responses, this is captured by a single scalar, the binary logit.) We consider several models:

Affine Multiplies each query embedding with the sign of the corresponding response (± 1 for positive/negative responses), sums the results, and passes the result to an affine layer.

Recurrent Applies a recurrent neural network (RNN) (Rumelhart et al., 1986; Werbos, 1988), specifically the Gated Recurrent Unit (GRU) (Cho et al., 2014), since it is commonly used in the literature and has competitive performance (Yang et al., 2020). RNNs are permutation-sensitive and can learn temporal dependencies, if they exist.

DeepSets Zaheer et al. (2017) introduced DeepSets, a special case of Janossy pooling (Murphy et al., 2018). This architecture applies an encoder to each element of the input (in our case, a sequence of pairs of a query-embedding and response), aggregates the results through a pooling operation (such as summation), and applies a decoder to the result. For the encoder and decoder, we use feedforward networks.

Attention Concatenates each query embedding with its corresponding response and applies multi-head self-attention (Vaswani et al., 2017).

Multiset Converts the query-response history to a $\mathbb{N}^{|\mathcal{Q}|\times|\mathcal{R}|}$ table of *counts* for each possible query-response pair, flattens it, and passes it to a feedforward network.

6. Results

We now describe our experimental results. For the models, we use a hidden layer size of 256 and four heads for the attention model. We use the Radam optimizer (Liu et al., 2019) with a learning rate of 10^{-5}, and a weight decay parameter of 10^{-3}. For training, we use a validation set split of 0.1, 100 epochs, a batch size of 32, and 20 trials.

First, we show the performance of various models in learning the response and utility functions from the synthetic dataset. Figure 1 shows the utility and response loss on the validation set for various architectures over the course of training. Solid lines indicate the mean across trials. Bands indicate a 0.95 confidence interval for this mean. The latter is computed using bootstrapping (Efron, 1979), specifically the bias-corrected and accelerated (BCa) method (Efron, 1987).

Second, we combine the best-performing model with different elicitation algorithms (e.g., those based on DLS or MCTS) to create an elicitation *policy*. The user response model is used at the search tree's chance nodes, and the utility model is used at its leaves. We evaluate the resulting policy by deploying it with the synthetic users, generating new episodes, and recording the resulting episode utilities. Results are shown in Figure 3. We include different planning depths for DLS and simulation budgets for MCTS. Dots show means across trials, while error bars indicate a 0.95 confidence interval for this mean. Our results show that planning-based policies yield better performance.

7. Conclusion

In this work, we propose a model-free approach to preference elicitation. It avoids simplistic, restrictive modeling assumptions and instead leverages function approximation to learn the

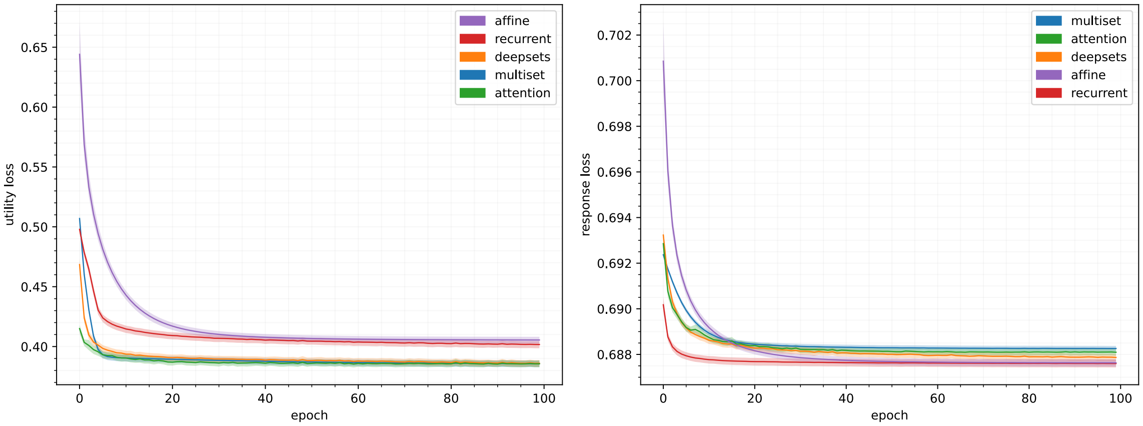

Figure 1: Utility and response loss on the validation set for various architectures.

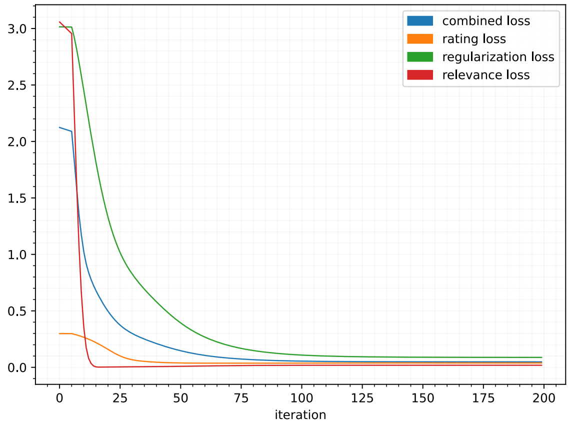

Figure 2: Embedding loss.

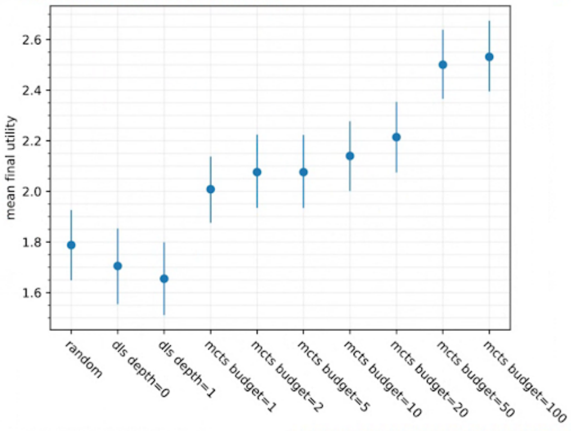

Figure 3: Episode utility.

quantities needed for PE. We explore multiple architectures and planning algorithms and demonstrate improvements in recommendation quality with respect to a natural baseline.

In future work, we intend to explore additional datasets and elicitation policies. Another important direction is training in an online fashion (like AlphaZero and MuZero) with simulated users, to learn more accurate value/policy functions. Finally, we hope to test an approach that first learns an environment dynamics model (user responses and item recommendation utilities) from offline data, and then runs *online* reinforcement learning against this learned model. We speculate that it may be useful to learn an *ensemble* of such models (or an uncertainty-aware model, such as a Bayesian neural network) to prevent overfitting the policy to a learned model that is different from the true environment.

Acknowledgments

We thank Chih-wei Hsu and Yinlam Chow from Google Research for useful discussions.

References

Ali Abbas. Entropy methods for adaptive utility elicitation. *IEEE Transactions on Systems, Science and Cybernetics*, 34(2):169–178, 2004.

Fabian Abel, Qi Gao, Geert-Jan Houben, and Ke Tao. Analyzing user modeling on Twitter for personalized news recommendations. In *User Modeling, Adaption and Personalization*, 2011.

Ioannis Antonoglou, Julian Schrittwieser, Sherjil Ozair, Thomas K Hubert, and David Silver. Planning in stochastic environments with a learned model. In *International Conference on Learning Representations*, 2022.

Igor Babuschkin, Kate Baumli, Alison Bell, Surya Bhupatiraju, Jake Bruce, Peter Buchlovsky, David Budden, Trevor Cai, Aidan Clark, Ivo Danihelka, Antoine Dedieu, Claudio Fantacci, Jonathan Godwin, Chris Jones, Ross Hemsley, Tom Hennigan, Matteo Hessel, Shaobo Hou, Steven Kapturowski, Thomas Keck, Iurii Kemaev, Michael King, Markus Kunesch, Lena Martens, Hamza Merzic, Vladimir Mikulik, Tamara Norman, George Papamakarios, John Quan, Roman Ring, Francisco Ruiz, Alvaro Sanchez, Rosalia Schneider, Eren Sezener, Stephen Spencer, Srivatsan Srinivasan, Wojciech Stokowiec, Luyu Wang, Guangyao Zhou, and Fabio Viola. The DeepMind JAX Ecosystem, 2020.

JesúS Bobadilla, Fernando Ortega, Antonio Hernando, and Jesús Bernal. A collaborative filtering approach to mitigate the new user cold start problem. *Knowledge-based systems*, 26:225–238, 2012.

Nadjet Bourdache, Patrice Perny, and Olivier Spanjaard. Incremental elicitation of rank-dependent aggregation functions based on bayesian linear regression. In *Proceedings of the Twenty-eighth International Joint Conference on Artificial Intelligence (IJCAI-19)*, pages 2023–2029, 2019.

Craig Boutilier. A POMDP formulation of preference elicitation problems. In *Proceedings of the Eighteenth National Conference on Artificial Intelligence (AAAI-02)*, pages 239–246, 2002.

Craig Boutilier. Computational decision support: Regret-based models for optimization and preference elicitation. In P. H. Crowley and T. R. Zentall, editors, *Comparative Decision Making: Analysis and Support Across Disciplines and Applications*, pages 423–453. Oxford University Press, Oxford, 2013.

Craig Boutilier, Relu Patrascu, Pascal Poupart, and Dale Schuurmans. Constraint-based optimization and utility elicitation using the minimax decision criterion. *Artifical Intelligence*, 170(8–9):686–713, 2006.

Darius Braziunas and Craig Boutilier. Assessing regret-based preference elicitation with the UTPREF recommendation system. In *Proceedings of the Eleventh ACM Conference on Electronic Commerce (EC'10)*, pages 219–228, Cambridge, MA, 2010.

Gregory Canal, Andy Massimino, Mark Davenport, and Christopher Rozell. Active embedding search via noisy paired comparisons. In *International Conference on Machine Learning*, pages 902–911, 2019.

Urszula Chajewska, Daphne Koller, and Ronald Parr. Making rational decisions using adaptive utility elicitation. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence (AAAI-00)*, pages 363–369, 2000.

Kyunghyun Cho, Bart van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: encoder-decoder approaches. In *SSST-8*, pages 103–111, 2014.

Rémi Coulom. Efficient selectivity and backup operators in Monte-Carlo tree search. In *Computers and Games*, 2007.

Paul Covington, Jay Adams, and Emre Sargin. Deep neural networks for YouTube recommendations. In *Proceedings of the 10th ACM Conference on Recommender Systems (RecSys16)*, pages 191–198, Boston, 2016.

Ivo Danihelka, Arthur Guez, Julian Schrittwieser, and David Silver. Policy improvement by planning with Gumbel. In *International Conference on Learning Representations (ICLR)*, 2022.

Bradley Efron. Bootstrap methods: Another look at the jackknife. *The Annals of Statistics*, 7:1–26, 1979.

Bradley Efron. Better bootstrap confidence intervals. *Journal of the American Statistical Association*, 82:171–185, 1987.

Gregory Farquhar, Tim Rocktaeschel, Maximilian Igl, and Shimon Whiteson. TreeQN and ATreeC: Differentiable tree planning for deep reinforcement learning. In *International Conference on Learning Representations (ICLR)*, 2018.

Arthur Guez, Theophane Weber, Ioannis Antonoglou, Karen Simonyan, Oriol Vinyals, Daan Wierstra, Remi Munos, and David Silver. Learning to search with MCTSnets. In *International Conference on Machine Learning*, 2018.

Arthur Guez, Mehdi Mirza, Karol Gregor, Rishabh Kabra, Sébastien Racanière, Théophane Weber, David Raposo, Adam Santoro, Laurent Orseau, Tom Eccles, et al. An investigation of model-free planning. In *International Conference on Machine Learning*, pages 2464–2473. PMLR, 2019.

Shengbo Guo and Scott Sanner. Real-time multiattribute Bayesian preference elicitation with pairwise comparison queries. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 289–296, 2010.

Blake Hallinan and Ted Striphas. Recommended for you: The Netflix prize and the production of algorithmic culture. *New Media & Society*, 18:117–137, 2016.

F. Maxwell Harper and Joseph A. Konstan. The MovieLens datasets: History and context. *ACM Transactions on Interactive Intelligent Systems*, 5(4):19:1–19:19, 2016.

Hillary A. Holloway and III Chelsea C. White. Question selection for multiattribute decision-aiding. *European Journal of Operational Research*, 148:525–543, 2003.

Ronald A. Howard and James E. Matheson, editors. *Readings on the Principles and Applications of Decision Analysis.* Strategic Decision Group, Menlo Park, CA, 1984.

Vijay S. Iyengar, Jon Lee, and Murray Campbell. Q-Eval: Evaluating multiple attribute items using queries. In *Proceedings of the Third ACM Conference on Electronic Commerce (EC'01)*, pages 144–153, Tampa, FL, 2001.

Ralph L. Keeney and Howard Raiffa. *Decisions with Multiple Objectives: Preferences and Value Trade-offs.* Wiley, New York, 1976.

Xuan Nhat Lam, Thuc Vu, Trong Duc Le, and Anh Duc Duong. Addressing cold-start problem in recommendation systems. In *Proceedings of the 2nd International Conference on Ubiquitous Information Management and Communication*, 2008.

Greg Linden, Brent Smith, and Jeremy York. Amazon.com recommendations: Item-to-item collaborative filtering. *IEEE Distributed Systems Online*, 4, 2003.

Liyuan Liu, Haoming Jiang, Pengcheng He, Weizhu Chen, Xiaodong Liu, Jianfeng Gao, and Jiawei Han. On the variance of the adaptive learning rate and beyond. *arXiv:1908.03265*, 2019.

Ryan L. Murphy, Balasubramaniam Srinivasan, Vinayak Rao, and Bruno Ribeiro. Janossy pooling: Learning deep permutation-invariant functions for variable-size inputs. *arXiv:1811.01900*, 2018.

Junhyuk Oh, Satinder Singh, and Honglak Lee. Value prediction network. In *Advances in Neural Information Processing Systems*, 2017.

Aditya Pal, Chantat Eksombatchai, Yitong Zhou, Bo Zhao, Charles Rosenberg, and Jure Leskovec. PinnerSage: Multi-modal user embedding framework for recommendations at Pinterest. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2020.

Al Mamunur Rashid, George Karypis, and John Riedl. Learning preferences of new users in recommender systems: An information theoretic approach. *SIGKDD Explor. Newsl.*, 10:90–100, 2008.

Lior Rokach and Slava Kisilevich. Initial profile generation in recommender systems using pairwise comparison. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 42:1854–1859, 2012.

David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Learning representations by back-propagating errors. *Nature*, 1986.

Ruslan Salakhutdinov and Andriy Mnih. Probabilistic matrix factorization. In *Advances in Neural Information Processing Systems 20 (NIPS-07)*, pages 1257–1264, 2007.

Ahti A. Salo and Raimo P. Hamalainen. Preference ratios in multiattribute evaluation (PRIME)-elicitation and decision procedures under incomplete information. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 31:533–545, 2001.

Julian Schrittwieser, Ioannis Antonoglou, Thomas Hubert, Karen Simonyan, Laurent Sifre, Simon Schmitt, Arthur Guez, Edward Lockhart, Demis Hassabis, Thore Graepel, et al. Mastering Atari, Go, chess and shogi by planning with a learned model. *Nature*, 588: 604–609, 2020.

David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of Go with deep neural networks and tree search. *nature*, 529:484–489, 2016a.

David Silver, Aja Huang, Chris J. Maddison, Arthur Guez, Laurent Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel, and Demis Hassabis. Mastering the game of Go with deep neural networks and tree search. *Nature*, 2016b.

David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, Yutian Chen, Timothy Lillicrap, Fan Hui, Laurent Sifre, George van den Driessche, Thore Graepel, and Demis Hassabis. Mastering the game of Go without human knowledge. *Nature*, 2017.

David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, Timothy Lillicrap, Karen Simonyan, and Demis Hassabis. A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play. *Science*, 2018.

Aviv Tamar, YI WU, Garrett Thomas, Sergey Levine, and Pieter Abbeel. Value iteration networks. In *Advances in Neural Information Processing Systems*, 2016.

Olivier Toubia, Duncan I. Simester, John R. Hauser, and Ely Dahan. Fast polyhedral adaptive conjoint estimation. *Marketing Science*, 22(3):273–303, 2003.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems 30 (NIPS-17)*, 30, 2017.

Ivan Vendrov, Tyler Lu, Qingqing Huang, and Craig Boutilier. Gradient-based optimization for bayesian preference elicitation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34:10292–10301, 2020.

Paolo Viappiani and Craig Boutilier. Optimal Bayesian recommendation sets and myopically optimal choice query sets. In *Advances in Neural Information Processing Systems 23 (NIPS)*, pages 2352–2360, 2010.

Paul J. Werbos. Generalization of backpropagation with application to a recurrent gas market model. *Neural Networks*, 1988.

Shudong Yang, Xueying Yu, and Ying Zhou. LSTM and GRU neural network performance comparison study: Taking yelp review dataset as an example. In *2020 International Workshop on Electronic Communication and Artificial Intelligence (IWECAI)*, pages 98–101, 2020.

Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Russ R Salakhutdinov, and Alexander J Smola. Deep sets. *Advances in Neural Information Processing Systems 30 (NIPS-17)*, 30, 2017.

Zhibing Zhao, Haoming Li, Junming Wang, Jeffrey Kephart, Nicholas Mattei, Hui Su, and Lirong Xia. A cost-effective framework for preference elicitation and aggregation. *arXiv:1805.05287*, 2018.