A computational model for decision tree search

Bas van Opheusden, Gianni Galbiati, Zahy Bnaya, Yunqi Li & Wei Ji Ma
Center for Neural Science & Department of Psychology, New York University
{basvanopheusden,gianni.galbiati,zahy.bnaya,yunqi.li,weijima}@nyu.edu

Abstract

How do people plan ahead in sequential decision-making tasks? In this article, we compare computational models of human behavior in a challenging variant of tic-tac-toe, to investigate the cognitive processes underlying sequential planning. We validate the most successful model by predicting choices during games, two-alternative forced choices and board evaluations. We then use this model to study individual skill differences, the effects of time pressure and the nature of expertise. Our findings suggest that people perform less tree search under time pressure, and that players search more as they improve during learning.

Keywords: Sequential decision-making, Behavioral modeling, Expertise

Introduction

Imagine you are deciding if you should run for President of the United States in 2020. To make that choice, you have to consider a sequence of future decisions. Will you run as a Republican, Democrat or Independent? If Democrat, will you run as a moderate or progressive candidate? What positions will you take on abortion or gun control? How will you distinguish yourself during the primaries? What line of attack will you choose in the Presidential Debates? You face a sequence of decisions, which together determine your electoral success. In short, you have to explore a decision tree.

Although the computations underlying human decision-making are extensively studied, the process by which people explore decision trees is less understood. Most work focuses on the neural implementation of learning and decision-making in small decision trees (Solway & Botvinick, 2015; Simon & Daw, 2011). However, with more choices and more available options, the decision tree grows exponentially, and people need to prune the tree (Huys et al., 2012).

There exists a large literature exploring human decision-making in chess, starting with de Groot's seminal article (A. D. de Groot, 1946). One central question in this literature is whether the superior performance of experts relies primarily on enhanced pattern recognition (Chase & Simon, 1973), increased tree search (Holding, 1985), or both. The relation between tree search and expertise is especially controversial, with both positive (Campitelli & Gobet, 2004) and negative (A. D. de Groot, 1946) results.

In this article, we investigate sequential decision-making in a two-player board game, which is much simpler than chess, but much more complex than traditional decision-making tasks. We develop a computational model that predicts people's choices on individual trials, and fit this model to data from individual participants. We then ask whether the computations performed by our model mimic the process by which people arrive at their decisions. Finally, we use our model to investigate the nature of expertise in our game.

Experiments

Task. To investigate the computations underlying sequential decision-making, we collected data from people playing a variant of tic-tac-toe, in which players need to make 4-in-a-row on a 4-by-9 board (figure 1A). Despite these simple rules, the game is surprisingly challenging and fun to play. Because the game is deterministic without hidden information, it is theoretically solvable. Using alpha-beta pruning and threat tree search (Allis et al., 1994), we were able to derive a weak solution: the first player can force a win by opening on the central square. However, with perfect defense, the second player can delay the win for 17 moves.

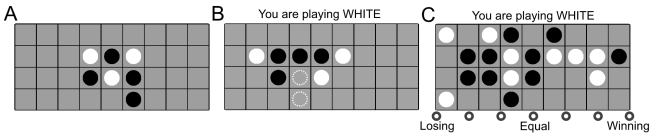

Figure 1: Task. **A.** Two players take turns placing black or white pieces on a 4-by-9 board, and the first to achieve 4-in-a-row (horizontal, diagonal or vertical), wins the game. **B.** In the 2AFC task, participants see a board and two candidate moves, and indicate their preferences. **C.** In the evaluation task, participants see a board position and report their estimated winning chances on a 7-point scale.

Participants. We conducted four experiments: human-vs-human ($N = 40$ participants), generalization ($N = 40$), time pressure ($N = 30$), and learning ($N = 30$). We recruited participants through the NYU psychology research participant system, flyers, a sign-up link on our lab webpage or personal communication. We did not collect demographic data. We compensated participants 12 per hour, but did not incentivize task performance.

Procedure. In the human-vs-human experiment, we divided participants into pairs. Participants in each pair played games against each other without time constraints for 50 minutes, switching colors every game. In the generalization experiment, participants performed three tasks: playing the game against a computer opponent for 30 minutes, 82 trials of a two-alternative forced-choice (2AFC) between moves in a given board position (figure 1B), and 82 board evaluation trials, in which they rated their winning chances in given board positions on a 7-point scale (figure 1C). The time pressure experiment was identical to the human-vs-computer component of the generalization experiment, except that for each game, we added a time limit randomly selected between 5, 10 or 20 seconds per move. If participants exceeded the time

limit, they lost the game. The learning experiment consisted of 5 sessions, no more than 3 days apart. In sessions 1, 3 & 5, participants played against computers for 30 minutes, then completed 60 trials each of the 2AFC and evaluation tasks. In session 2 & 4, they played against computers for the entire 50-minute session.

In all human-vs-computer games, the computer opponents implemented an early version of our computational model for people's decision-making process, with parameters adapted from fits on human-vs-human games. We created 30 AI agents, grouped by playing strength into 6 groups of 5 agents each, and matched participants with AI opponents through a one-up, one-down staircase procedure.

In the 2AFC and evaluation task, each participant completed the same trials in shuffled order. We selected board positions and move options that maximize an approximation to mutual information between model parameters and move choice, in order to present participants with interesting and informative choices.

Model

Value function. The core component of our model is an evaluation function $V(s)$ which assigns values to board states s. We use a weighted linear sum of 5 features: center, connected 2-in-a-row, unconnected 2-in-a-row, 3-in-a-row and 4-in-a-row. The center feature assigns a value to each square, and sums up the values of all squares occupied by the player's pieces. This value of each square is inversely proportional to its Euclidean distance from the board center. The other features count how often particular patterns occur on the board (horizontally, vertically, or diagonally):

Connected 2-in-a-row: two adjacent pieces with enough empty squares around them to complete 4-in-a-row.

Unconnected 2-in-a-row: two non-adjacent pieces which lie on a line of four contiguous squares, with the remaining two squares empty.

3-in-a-row: three pieces which lie on a line of four contiguous squares, with the remaining square empty. This pattern represents an immediate winning threat.

4-in-a-row: four pieces in a row. This pattern appears only in board states where a player has already won the game.

We associate weights w_i to these features, and write

$$V(s) = c_{\text{self}} \sum_{i=0}^{4} w_i f_i(s, \text{self}) - c_{\text{opp}} \sum_{i=0}^{4} w_i f_i(s, \text{opponent})$$

where $c_{\text{self}} = C$ and $c_{\text{opp}} = 1$ whenever the player is to move in state s, and $c_{\text{self}} = 1$ and $c_{\text{opp}} = C$ when it is the opponent's move. The scaling constant C captures value differences between "active" and "passive" features. For example, a three-in-a-row feature signals an immediate win on the player's own move, but not the opponent's.

Tree search. The evaluation function guides the construction of a decision tree with an iterative best-first search algorithm. Each iteration, the algorithm chooses a board position to explore further, evaluates the positions resulting from each legal move, and prunes all moves with value below that of the best move minus a threshold. After each iteration, the algorithm stops with a probability γ, resulting in a geometric distribution over the total number of iterations.

Noise. To account for variability in people's choices, we add three sources of noise. Before constructing the decision tree, we randomly drop features (at specific locations and orientations), which are omitted during the calculation of $V(s)$ anywhere in the tree. During tree search, we add Gaussian noise to $V(s)$ in each node. Finally, we include a lapse rate λ.

The components of our computational model are inspired by behavioral studies of human decision-making. Tree search, as a mechanism whereby people mentally simulate the consequences of available actions, is similar to "level-K reasoning" (Arad & Rubinstein, 2012) in behavioral economics. In other decision-making tasks, people have been shown to prune away options leading to immediate losses but long-term gains (Huys et al., 2012). Feature dropping reflects shift in endogenous attention (to spatial locations, orientation or feature types), corresponding to participants overlooking relevant features on the board. Finally, feature-based evaluation functions, value noise and lapse rates are all common in reinforcement learning.

There also exists neural evidence consistent with our model. In rats, dynamic search and exploration of possible paths at junctions in a T-maze have been linked to preplay sequences in hippocampal place cells (Johnson & Redish, 2007). In humans, tree search is associated with neural activity in the ventral striatum (Simon & Daw, 2011) and ventromedial prefrontal cortex (Lee, Shimojo, & ODoherty, 2014).

Methods

Estimating task performance. To quantify task performance in human-vs-computer games, we use the Elo rating system (Elo, 1978), which estimates playing strength from game results, independent of the moves played. We append the results of games from all 4 experiments to a computer-vs-computer tournament, and estimate ratings jointly for all humans and computers with a Bayesian optimization algorithm (Hunter, 2004). To calculate performance in the 2AFC task, we calculate the agreement between a participant's choices and those of an optimal agent with random tie-breaking. In the evaluation task, we define performance as the correlation between a participant's choices and the optimal rankings.

Estimating model parameters The model has 10 parameters: the 5 feature weights, the active-passive scaling constant C, the pruning threshold, stopping probability γ, feature drop rate δ and the lapse rate λ. We infer these parameters for individual participants and individual learning sessions or time limit conditions with maximum-likelihood estimation. We estimate the log probability of a participant's move in a given board position with inverse binomial sampling (M. H. de Groot, 1959), and optimize the log-likelihood function with multilevel coordinate search (Huyer & Neu-

maier, 1999). We account for potential overfitting by reporting 5-fold cross-validated log-likelihoods, with the same testing-training splits for all models.

Model comparison

To test how well our model predict participants' choices, we compare its log-likelihood on human-vs-human games to that of 25 alternative models (figure 2). We test four categories of alternative models: lesions, generated by removing model components; extensions, generated by adding new model components; modifications, generated by replacing a model component with a similar implementation; and controls, which are structurally different from the main model.

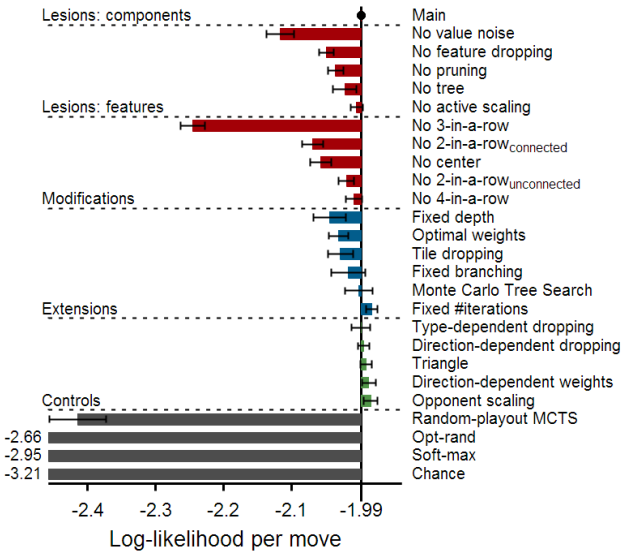

Figure 2: Cross-validated log-likelihood/move for our main model and 25 alternatives on the human-vs-human data. The bars show mean and s.e.m. across participants ($N = 40$). The main model fits better than lesions, most controls and some modifications, and approximately equally good as extensions or some other modifications.

Lesions. We create lesion models by forcing either one of the feature weights to zero, or removing the feature dropping, pruning, value noise, active-passive scaling or the entire decision tree. The no-tree model evaluates the positions after each possible move, and chooses the one with maximum value. It contains feature dropping and value noise but no pruning.

Extensions. We consider extending the model with a feature that recognizes to a three-piece pattern arranged in a triangle, or multiplying the weights for diagonally and vertically oriented features by scaling constants c_{diag} or c_{vert}, respectively. Alternatively, we extend the main model by allowing feature drop rates to differ between features of different types (2-in-a-row, 3-in-a-row, etc) or orientations. Finally, we test a model in which all weights for the opponent's features are scaled by a factor c_{opp}, which thereby controls the balance between attack and defense.

Modifications. We modify the model by fixing the number of iterations of the search algorithm to a constant instead of the geometric distribution prescribed by the main model. Alternatively, we amend the search process to explore each branch of the tree up to fixed depth, or the pruning rule to keep only the K best moves (according to the evaluation function), where the branching factor K is again fixed. For a more drastic modification, Monte Carlo Tree Search (MCTS) estimates state values not by calling the evaluation function $V(s)$, but by aggregating outcomes of simulated games between no-tree agents. It also extends the best-first search algorithm by adding a term that favors exploration (investigating unexplored moves) over exploitation (further investigating already explored moves). We consider fixing the feature weights to the optimal solution, i.e. those weights that maximize the correlation between $V(s)$ and the game-theoretic value of the position s. Finally, we modify the attention mechanism from dropping random features from the evaluation function to dropping random branches from the decision tree.

Controls. We consider MCTS with completely random playouts, or a mixture model between optimal and random play. The optimal agent enumerates all candidate moves that preserve the game-theoretic value of the position, and chooses randomly between them. Another control model, labeled soft-max, assigns a value to each square on the board (enforced to obey reflection/rotation symmetry), and chooses a move with a softmax decision rule, constrained to unoccupied squares.

All lesioned models fit worse than the full model. The most impactful lesions are specific features (3-in-a-row, connected 2-in-a-row and center) and sources of variability (value noise and feature dropping). Lesioning the pruning mechanism or the entire tree search algorithm has a less dramatic effect, which can be partially explained by parameter trade-offs. Finally, some lesions (active-passive scaling, unconnected 2-in-a-row and 4-in-a-row) cause only small reductions in log-likelihood. Most modifications also worsen the main model, but the Monte Carlo Tree Search model is equally good and the "fixed iterations" model slightly outperforms it. The model extensions also slightly increase the main model's performance. Finally, all control models fit much worse than the main model.

Unfortunately, the model comparison does not reveal a unique best-fitting model, meaning that we did not collect enough data to determine precise details of people's thought process. For example, we cannot distinguish between tree search algorithms (best-first search or MCTS) or determine specifics of the best-first search algorithm (pruning and number of iterations). Alternatively, different participants may use different strategies. However, the model comparison does suggest that any model that can predict human choices needs to contain a feature-based evaluation function, and mechanisms for attentional oversights and tree search.

Generalization to 2AFC and Evaluation

Next, we show that the model can generalize by estimating parameters from subjects' choices in games against computers and predicting their choices in the 2AFC or evaluation tasks with minimal additional assumptions. To select an option on a 2AFC trial, the only change we make is to initialize the tree search algorithm with a three-node decision tree with the current board position as the initial node and the two available candidate moves as children. On an evaluation trial, we execute the tree search algorithm as usual, then measure the value of the root node. We then convert this value to a seven-point scale by transforming $v \to 3 + 4\tanh(v/20)$ and rounding to the nearest integer.

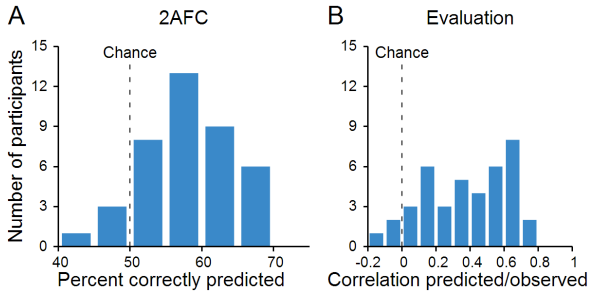

Figure 3: **A.** Histogram of the percentage of correctly predicted 2AFC choices by our model across $N = 40$ participants. We fit parameters for each participant on their choices in games against computers. The dashed line indicates the accuracy of a random prediction. **B.** Same for the evaluation task, where we quantify goodness-of-fit as the correlation across trials between rankings predicted by the model and reported by a participant.

The average accuracy of the model prediction on 2AFC data is $58.6 \pm 1.0\%$ (figure 3A), the average correlation between predicted and observed evaluations is $\rho = 0.38 \pm 0.04$ (figure 3B). The prediction is better than chance for 36/40 participants in the 2AFC task, and 37/40 for evaluation.

Even though our model predicts participants' choices in these additional tasks well on average, the goodness-of-fit is highly variable across participants. This variability in goodness-of-fit is correlated across subjects between the three tasks (2AFC-evaluation: $\rho = 0.54$, $p < 0.001$; 2AFC-games: $\rho = 0.35$, $p < 0.05$; evaluation-games: $\rho = 0.24$, $p = 0.12$). Moreover, on the 2AFC and evaluation task, goodness-of-fit correlates with participants' objective task performance (2AFC: $\rho = 0.56$, $p < 0.001$; evaluation: $\rho = 0.96$, $p < 0.001$). This suggests that the variability in goodness-of-fit can at least partially be explained by differences in intrinsic variability across participants.

How experimental manipulations affect model parameters

To further support the model, we investigate whether its parameters respond in predictable ways to experimental manipulations. As our first manipulation, we introduce time constraints of 5, 10 or 20 seconds per move. Second, we conduct an experiment in which participants play the game for 5 sessions.

Given a set of parameters for an individual participant in a time limit condition or learning session, we simulate moves made by the model in a database of pre-determined positions and measure 3 statistics of its process: the percentage of dropped features, the value quality (correlation between $V(s)$ and the game-theoretic value $V^*(s)$) and the mean tree size (number of nodes in its decision tree). Note that tree size incorporates both the width and depth of the decision tree.

Based on the literature on expertise and time pressure in chess, we expected that time constraints would reduce tree size but not affect value function quality. In the learning experiment, we expected the value function quality to increase across sessions and the tree size to remain constant or increase only slightly. Since chess algorithms often do not explicitly include feature dropping or similar mechanisms, we made no predictions for its trajectory. Finally, we predict that experience increases participants' task performance while time pressure reduces it.

Time pressure To test the effectiveness of time constraints to manipulate participants' behavior, we first plot the distribution of response times in the three conditions, as well as the response times from the unconstrained (generalization) experiment (figure 4A). Adding time pressure causes an overall shift in the response time distribution regardless of the time limit. Additionally, participants play faster with shorter time constraints. Surprisingly, there is no consistent effect of time constraints on participants' performance (figure 4B).

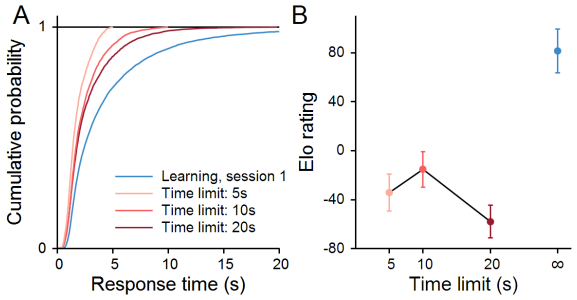

Figure 4: **A.** Empirical cdf of response times in the three conditions of the time pressure experiment (red), and the generalization experiment (blue). In the latter experiment, players could take arbitrary amounts of time, which we denote as an infinite time limit. People play faster with shorter time limits. **B.** Task performance, quantified by Elo rating, for the same experiments and conditions. Error bars indicate mean and s.e.m. across participants ($N = 30$). The effect of time limits on performance is unclear.

In figure 5 (top), we show the feature drop rate, value function quality and tree size in different time limit conditions. Compared to the unconstrained experiment, participants build

smaller trees and drop more features, while the value function quality is similar. The impact of the time constraint on tree size becomes larger with shorter time limits, but the feature drop rate shows the opposite trend and is at its highest in the 20-second condition. We speculate that the stress of potentially losing on time causes participants to pay more attention with shorter time limits, whereas with 20 seconds, they are more relaxed and make more attentional lapses.

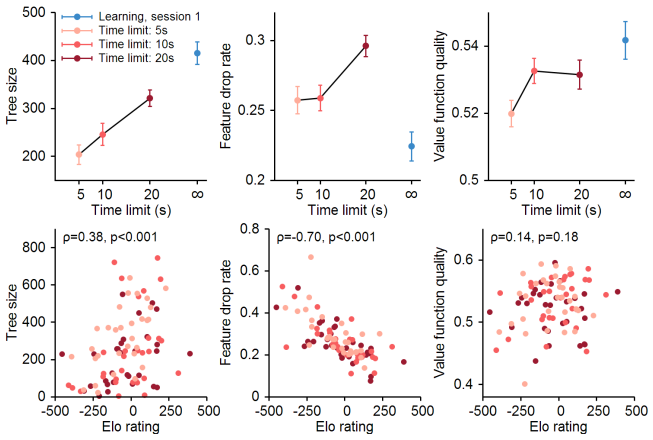

Figure 5: **Top row**. Estimated model parameters in the time pressure and generalization experiments. Error bars denote mean and s.e.m. across participants. The model infers a relation between time limit and tree size, but unclear effects on feature dropping and the value function quality. **Bottom row**. Model parameters and Elo rating for each participant in each time limit condition. The tree size and feature drop rate correlate with Elo rating, but value function quality does not.

To understand the surprising negative result of figure 4, we investigate how Elo rating and parameter estimates correlate across both individuals and time limit conditions (figure 5, bottom). Stronger players (in all time limit conditions) are estimated to build larger decision trees and drop fewer features. Therefore, the increased tree size with longer time limit predicts a performance increase, but the increased feature drop rate predicts decreased performance. These opposite effects happen to be approximately equal, which explains the lack of correlation between time limit and Elo rating.

Learning We first validate that experience affects participants' behavior by plotting Elo rating as a function of session number (figure 6). Next, we investigate changes in parameters across sessions (figure 7, top). Tree size increases across sessions, feature drop rate decreases and value function quality remains constant. As in the time pressure experiment, tree size and feature drop rate correlate with Elo rating on an individual level (figure 7, bottom), and the change in parameter estimates across sessions explains changes in task performance. Experienced players build larger decision trees and drop fewer features, both of which predict increased playing strength, which matches the data.

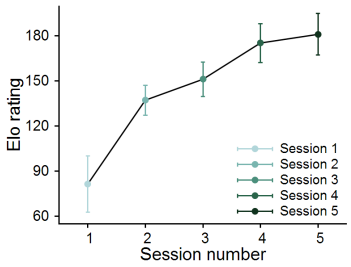

Figure 6: Elo rating of $N = 30$ participants in the learning experiment (mean and s.e.m. across participants). As participants gain expertise, they play stronger.

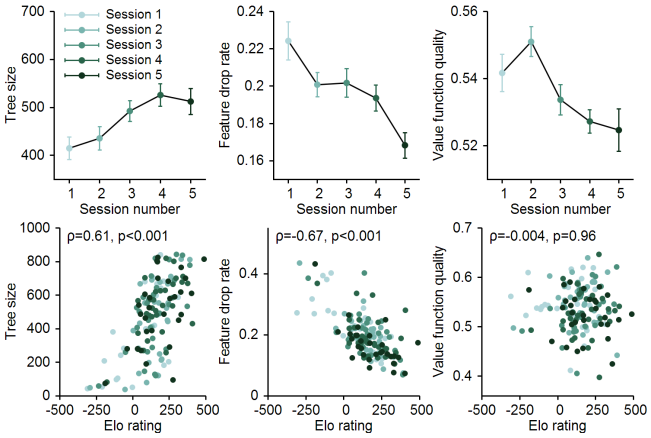

Figure 7: **Top**: Model parameters as a function of sessions completed in the learning experiment. Over the course of learning, tree size is estimated to increase while feature dropping decreases. The value function quality decreases, but only slightly. **Bottom**: Model parameters and Elo ratings for each participant in each session of the learning experiment. Both tree size and feature dropping correlate with Elo, but value function quality does not.

Discussion

Limitations. Our model has three conceptual limitations. First, although its parameters shift as participants acquire expertise, the model does not describe how these shifts arise from their experience (their specific move choices and rewards). Instead, model parameters are stationary within each session. Moreover, because model parameters are constant while participants play against multiple AI opponents per session, the model cannot capture strategic adaptations based on an opponent's game play. Finally, the model assumes that people make decisions independently on every move, ignoring potential long-term planning or caching of partial game trees between moves. We make these assumptions out of necessity, because parameter inference is already challenging.

Relation with chess literature. Contrary to the chess literature, in which the superior pattern recognition of chess experts is evident from board reconstruction experi-

ments (Chase & Simon, 1973) and eye movements (Reingold, Charness, Pomplun, & Stampe, 2001), we find no changes in value function quality with expertise or individual skill differences. Stronger players might use features outside our model space, and the lack of correlation could be a false negative. Alternatively, perhaps chess and 4-in-a-row are qualitatively different domains of expertise. Chess contains many non-obvious features (pawn structure, the bishop pair) or non-obvious feature weights (bishops and knights are equally strong). By contrast, in our task, people's intuitive priors (three-in-a-row is good) happen to be correct.

Our finding of increased tree search with longer time controls is consistent with chess studies that conceptualize pattern recognition and tree search as fast and slow processes, respectively (Chabris & Hearst, 2003). However, the strong dependence between expertise and tree search is unexpected. We first investigate whether this effect could have arisen from incorrect model assumptions. Specifically, players may use unmodeled features, stronger players may assign those features higher weights, and those feature weights may trade off with additional tree search in our model. However, by analyzing parameter estimates in lesion models, we find no such trade-offs. Therefore, our results reflect differences between 4-in-a-row and chess, or a methodological improvement. Conclusions about tree search in chess derive almost solely from verbal reports, whereas we use the more principled method of parameter inference in a behavioral model.

Conclusion

We built a computational model that predicts people's choices in a two-player board game. The model posits three computational principles for sequential decision-making: a feature-based evaluation function, attentional oversights and tree search. All three components are necessary to explain participants' behavior, but the data does not constrain details of their implementation such as the order by which nodes are visited during search, or how long the search process continues before players finalize their decision.

The model generalizes to predict choices in a two-alternative forced-choice task and a board evaluation task. This suggests that the model doesn't just fit a mapping from boards to moves, but that it captures aspects of the computational process that underlies decision-making in all three tasks. Furthermore, the feature drop rate and tree size change in predictable ways when we expose participants to manipulations in time pressure and experience. These changes account for participants' task performance, suggesting that these specific parameters reflect some task-relevant characteristic of participants' cognitive process. Furthermore, these two behavioral characteristics are dissociable, since in the time pressure experiment, both tree size and feature dropping increase across conditions, whereas in the learning experiment, tree size increases while feature dropping decreases. In the future, we aim to further support our model as a description of the computational process underlying people's move choices by using it to predict response times and eye movements.

Acknowledgments

This work was supported by grant IIS-1344256 from the National Science Foundation

References

Allis, L. V., et al. (1994). *Searching for solutions in games and artificial intelligence*. Ponsen & Looijen.

Arad, A., & Rubinstein, A. (2012). The 11–20 money request game: a level-k reasoning study. *The American Economic Review*, *102*(7), 3561–3573.

Campitelli, G., & Gobet, F. (2004). Adaptive expert decision making: Skilled chess players search more and deeper.

Chabris, C. F., & Hearst, E. S. (2003). Visualization, pattern recognition, and forward search: Effects of playing speed and sight of the position on grandmaster chess errors. *Cognitive Science*, *27*(4), 637–648.

Chase, W. G., & Simon, H. A. (1973). Perception in chess. *Cognitive psychology*, *4*(1), 55–81.

de Groot, A. D. (1946). *Het denken van den schaker: een experimenteel-psychologische studie*. Noord-Hollandsche Uitgevers Maatschappij.

de Groot, M. H. (1959). Unbiased sequential estimation for binomial populations. *The Annals of Mathematical Statistics*, 80–101.

Elo, A. E. (1978). *The rating of chessplayers, past and present*. Arco Pub.

Holding, D. H. (1985). *The psychology of chess skill*. Lawrence Erlbaum.

Hunter, D. R. (2004). Mm algorithms for generalized bradley-terry models. *Annals of Statistics*, 384–406.

Huyer, W., & Neumaier, A. (1999). Global optimization by multilevel coordinate search. *Journal of Global Optimization*, *14*(4), 331–355.

Huys, Q. J., Eshel, N., O'Nions, E., Sheridan, L., Dayan, P., & Roiser, J. P. (2012). Bonsai trees in your head: how the pavlovian system sculpts goal-directed choices by pruning decision trees. *PLoS Comput Biol*, *8*(3), e1002410.

Johnson, A., & Redish, A. D. (2007). Neural ensembles in ca3 transiently encode paths forward of the animal at a decision point. *Journal of Neuroscience*, *27*(45), 12176–12189.

Lee, S. W., Shimojo, S., & ODoherty, J. P. (2014). Neural computations underlying arbitration between model-based and model-free learning. *Neuron*, *81*(3), 687–699.

Reingold, E. M., Charness, N., Pomplun, M., & Stampe, D. M. (2001). Visual span in expert chess players: Evidence from eye movements. *Psychological Science*, *12*(1), 48–55.

Simon, D. A., & Daw, N. D. (2011). Neural correlates of forward planning in a spatial decision task in humans. *Journal of Neuroscience*, *31*(14), 5526–5539.

Solway, A., & Botvinick, M. M. (2015). Evidence integration in model-based tree search. *Proceedings of the National Academy of Sciences*, *112*(37), 11708–11713.