Human Decisions versus Optimal Algorithm in Bisection Search Game

Quilee Simeon (qsimeon@mit.edu)
Ian Miller (immiller@mit.edu)

Abstract

A critical aspect of human cognition is the ability to effectively query the environment for information. Humans are not perfect computing machines. Sometimes people make suboptimal local decisions but end up very close to an optimal global decision. Humans also make use of feedback or censors from their environment to make decisions and eventually converge to the solution of a problem. We will be investigating the divergence between a known ideal algorithm prescription and human decision-making in a simple task in order to better understand how humans make decisions. We have designed a simple "Guess the Number" game where participants attempt to use as few turns as possible to guess a secret integer in a predefined number interval. We find that humans are initially not strategic in their decisions but eventually converge to expected information gain maximizing agents as the game progres. Humans are less efficient at the task than the algorithm because of their inability to maximize expected information gain early on in the game.

Keywords: expected information gain; bisection search

Introduction

As humans, we are continually tasked with decisions. At any given moment, we are ingesting an astounding amount of raw data, processing it, and using it to inform these decisions. Should I wake up now? Should I go brush my teeth? What should I wear? What shall I have for breakfast? Do I have enough time to make eggs for breakfast? In the span of a few minutes, we do what seems like an exhausting amount of mental work to exist in the real world. Sometimes, we make good decisions. Sometimes, we make bad ones. Oftentimes, it's hard to tell. Our success as individuals, to the extent that we have any control over it, hinges on the quality (and the execution) of our decisions. More important decisions have a greater effect on our success than less important decisions. More important decisions are usually dependent on a large number of factors, whereas less important decisions are usually dependent on a smaller number of factors. How good are humans at the right decisions? How does accuracy change when decisions become harder (or more important)? How much of an effect does our accuracy in intermediate decisions have on the accuracy of the larger, global decision? Previous work by Tsividis et al. investigates the reasons for local suboptimal decisions. Specifically, humans tend to make decisions in order to optimize: 1) the maximization of novel information gain, and 2) the minimization of time and resources spent to get to that local decision. In this paper, we will expand on that investigation.

Humans are not perfect computing machines. We make mistakes. Is there a particular reason for these mistakes? Sometimes, we make suboptimal local decisions but end up very close to an optimal global decision. We will be investigating the delta between ideal algorithm prescription and human decision making in order to better understand how humans make decisions.

Bisection search task

In order to answer these questions, we designed in PsychoPy a simple game that we called "Guess the Number." The goal was to have people solve a simple problem for which there is an optimal algorithmic course of action. The task was to guess with as few tries as possible a random secret integer out of an interval given interactive feedback on whether your current guess was too high or too low after each turn. Binary search or bisection search is an efficient algorithm for finding this number whereby half the interval is thrown out after every turn and the agent guesses the midpoint of the remaining feasible interval. This algorithm has a worst case complexity of $O(log_2 N)$. This means that for a fixed interval/number line of size N, the algorithm starts by choosing the midpoint of the interval as its first guess. If that guess is too low/high, the algorithm recurses on the interval bounded below/above by the previous guess, respectively. Bisecting the remaining interval after every guess is guaranteed to find the secret number with no more than $log_2 N$ guesses.

We distributed the link to an online hosting of our "Guess the Number" game to random people, many of whom were unlikely to be familiar with binary/bisection search as an algorithm in computer science. There were two different experiments/versions of "Guess the Number" that were run and different participants played in both versions. The versions differed only in the size of the interval by varying the upper bound of the number interval: mid-sized interval $[1 \;\; 10^3]$, large interval $[1 \;\; 10^5]$. We ran these different versions to investigate the effect, if any, of the size of the number interval on people's performance.

The procedure for all experiments/game versions was the same. Participants were introduced to a welcome screen that introduced the concept of the game along with the number interval for the particular version being played and instructions on how to enter responses (Figure 1). Participants were encouraged to play efficiently by being told that a final game score would be computed using their speed of response as well as the number of guesses they make (Figure 2). A random secret number was drawn for every game and each person played the game to completion (i.e. until they guessed the secret number). The online interface recorded every number guessed by the participant until game termination as well as the times between guesses. (In theory, a score metric could have been calculated and shown to participants using their reaction times and number of guesses but we did not compute this as it was irrelevant to this study.)

Figure 1.a: The first instruction screen for the large interval version of "Guess the Number".

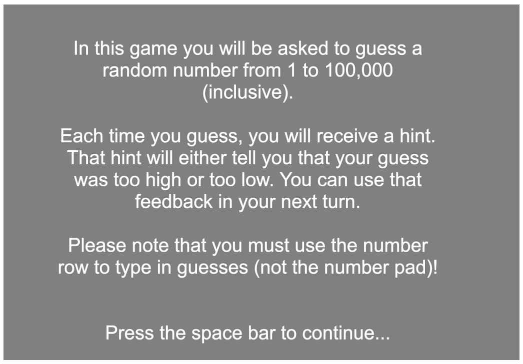
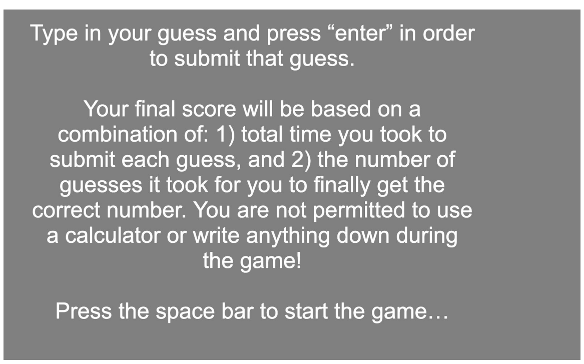


Figure 1.b: Second instruction screen for "Guess the Number" game. In reality, no such score is returned to participants at the end of the game. We included this detail so that people would be discouraged from taking their time.



The most inefficient approach to this problem/task is to guess randomly from the game interval on every turn. This approach makes no use of the information gained from previous guesses though the high/low interactive feedback of the game. On the other hand, because of the fact that the secret number is drawn (uniform) randomly from the number line, the optimal strategy is pure bisection search from the very start. This strategy is optimal from a frequentist perspective because if this game were played very many times for rewards inversely proportional to the number of guesses taken, pure bisection search would result in the largest expected return of any strategy. To show this we compared the number of guesses taken to find the secret number by humans and the bisection search algorithm.

We hypothesized that humans may not do perfect bisection search due to computing and time constraints, and perhaps even due to innate hypotheses about random number generation. We investigated this by comparing the distribution of first guesses of participants in the different versions of "Guess the Number" to pure bisection search, which will always choose the midpoint of the number interval as the first guess. Rather, at every turn people should seek to maximize the expected information gain (EIG) given their previous guesses which may not have been optimal. In the 'Model' section of this report, we argue that bisection search conditioned on previous guesses is the one-step EIG maximizing strategy. To efficiently guess the secret number participants should choose the midpoint of the smallest interval carved out by the feedback from previous guesses. That is, participants should guess the EIG maximizing value any given turn. The main analysis of this work is to investigate the divergence (what we call 'delta') between participant's actual guesses and the EIG maximizing guesses.

Model

We have adapted the expected information gain (EIG) model described in Tsividis et al (2014) as below.
Let I denote the number line/interval of the game. where L and U denote the $I = [L, U]$ global lower and upper bounds of the game interval, respectively. We ran two types of experiments or games, both with $L = 1$, and with U being some power of 10: mid-sized game ($U = 10^3$); large game ($U = 10^5$).

Let let $h \in I$ denote a hypothesis about the secret integer and let h^* be the true value of the secret integer. h^* is unknown to participants and is also randomly chosen at the start of every game: $h^* \sim Unif(L, U)$.

On each trial, participants choose an integer $a \in I$ and observe a ternary outcome d (*'eq'* if $a = h^*$; *'lo'* if $a < h^*$; and *'hi'* if $a > h^*$). If d = 'eq', then the participant has guessed the secret number and the game is over.

The posterior distribution over h is updated on each turn according to Bayes' rule:

$$P(h|d,a,D) \propto P(d|h,a) * P(h|D)$$

where D denotes the history of actions and outcomes prior to the current trial.

However, D can be summarized by just the interval $D = [l, u]$, where $l = max(\{1\} \cup \{previous\ guesses\ s.t.\ d = 'lo'\})$ and $u = min(\{U\} \cup \{previous\ guesses\ s.t.\ d = 'hi'\})$. For simplicity, we will denote the size of the interval *[l, u]* by $|D|$, where $|D| = u - l + 1$. (Note that $|D| <= |I|$.)

Therefore, the conditional prior $P(h|D) = \{1 / |D|$, if $h \in [l, u]$; 0, otherwise$\}$. To calculate the likelihood $P(d|h,a)$, we use the fact that h is an hypothesis about h^*, a is an individual's actual guess on a turn, and d is either *'hi'* or *'lo'* without terminating the game (d = *'eq'* ends the game):

$P(d='hi'|h,a) = P(a > h) = P(h < a) = (a - 1) / |D|$;
$P(d='lo'|h,a) = P(a < h) = P(h > a) = (|D| - a) / |D|$;
$P(d='eq'|h,a) = P(a == h) = P(h == a) = 1 / |D|$;

Therefore, the posterior distribution over h after each turn is:

$$P(h|d,a,D) = \sum h \in D(1 / |D|) * P(d|h,a) = P(d|h,a)$$

Intuitively, participants should choose actions that maximally reduce their uncertainty about the value of the hidden number. This corresponds to taking actions that maximally reduce posterior uncertainty, which can be quantified by the entropy:

$$H[P(h|d,a,D)] = -P(d|h,a) * logP(d|h,a)$$

Minimizing posterior entropy is equivalent to maximizing information gain (the reduction of entropy after choosing a and observing d:

$$IG(a,d) = H[P(h|D)] - H[P(h|d,a,D)]$$
$$= K + P(d|h,a) * logP(d|h,a)$$

where $K < 0$ represents the constant value of $H[P(h|D)] = -(1/|D|) * log(1 / |D|)$.

Because the outcome d is not available at the time of choosing $a \in I$, and considering only the outcomes of d that continue the game (*'hi'* or *'lo'*), the best that a participant can do at every turn is maximize expected information gain:

$$EIG(a) \propto \sum d \in \{\text{'hi', 'lo'}\} P(d|a,D) * IG(a,d)$$

where the posterior predictive distribution is given by:

$$P(d|a,D) = \sum h \in I\, P(d|h,a)P(h|a,D) = \sum h \in D\, P(d|h,a)P(h|D) = P(d|h,a)$$

The second equality is because h is conditionally independent of a and $P(h|D) = 0$ if $h \notin D$. The last equality uses $P(h|D) = 1 / |D|$ if $h \in D$.

Without showing rigorous proof, we argue that maximizing $EIG(a)$ is the discrete equivalent of minimizing the continuous function $x^2 * log(x) + (1-x)^2 * log(1-x)$ over the domain $0 <= x <= 1$. The solution is x=½ which is the midpoint of the domain (Figure 3). Analogously, the value of a^* that minimizes $EIG(a)$ is the midpoint of D.

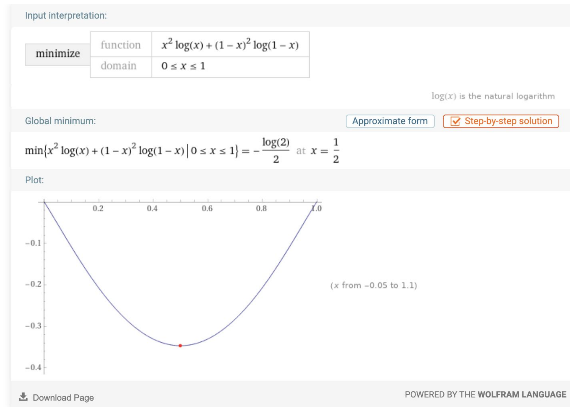


Figure 3: Screenshot of a WolframAlpha session solving the minimization problem argmin $x^2\log(x) + (1-x)^2\log(1-x)$, $x \in [0,1]$. This problem helps solve the EIG maximization problem.

This result confirms that bisection of the current feasible interval (i.e the interval over which the conditional prior $P(h|D)$ is non-zero) is the optimal strategy at every turn because it maximizes expected information gain (EIG). At any turn in the game, an agent should use the information from their prior guesses (specifically the points l and u as described previously) to make their next guess. The next guess should be $(u+l)//2$, the midpoint of D, where // represents integer division, or equivalently $floor((u+l)/ 2)$. For example, on the very first guess, $D = I = [L, U] = [1, U]$. The first guess of an EIG maximizing agent should therefore be $floor((U+1)/2)$. In the large interval game with $U = 10^3$, this would correspond to a first guess of 500. An agent that makes the EIG maximizing guess from the start of the game performs pure bisection search (Figure 4).

To help investigate how actual peoples' guesses deviate from the guesses expected of an EIG maximizing agent, we developed an algorithm that takes the actual guess sequence of participants and outputs the EIG maximizing guess sequence (Figure 5). We then represent the delta between actual human and EIG maximizing agent as the absolute difference at each step between the two sequences normalized by the size of the game interval.

Figure 4: Code snippet showing the optimal bisection search algorithm.

```python
# optimal bisection search
def bisection_search_global_optimal(secret, lower, upper):
    """
    Returns the optimal bisection search trace if
    information gain was maximized at every step.
    Args:
        secret (int): the secret number to be guessed.
        lower (int): inclusive lower bound of interval.
        upper (int): inclusive upper bound of interval.

    Returns:
        A list of the optimal bisection search trace.
    """
    trace = []
    guess = (upper + lower)//2
    while guess != secret:
        trace.append(guess)
        # update guess
        if guess < secret: # too low
            lower = guess
        elif guess > secret: # too high
            upper = guess
        guess = (upper + lower)//2
    # append last guess (which must be the secret number)
    trace.append(guess)
    return trace
```

Figure 5: Code snippet showing an algorithm for generating the EIG maximizing guess sequence of each participant.

```python
def bisection_search_conditional(secret, lower, upper, sequence):
    """
    Returns trace of normative optimal predictions for an agent that
    maximizes expected information gain at each turn.

    Args:
        secret (int): the secret number to be guessed.
        lower (int): inclusive lower bound of interval.
        upper (int): inclusive upper bound of interval.
        sequence (list): the guesses of an actual person.

    Returns:
        A list of the normative conditional bisection trace.

    """
    trace = [None]*len(sequence)

    for t in range(len(sequence)):
        if t > 0:

            # what the person guessed last turn
            pred = sequence[t-1]
            if pred < secret:
                # assumes person remembers lower
                lower = max(lower, pred)
            elif pred > secret:
                # assumes person remembers upper
                upper = min(upper, pred)

        # midpoint maximizes information
        guess = (upper + lower)//2
        trace[t] = guess

    return trace
```

Experiment 1: Mid-sized Interval

Method

Participants. 43 participants completed the experiment hosted online with Pavlovia.

Materials. We used the interval [1, 1000]. The secret number *h** was chosen uniformly at random at the start of the experiment for each participant.

Procedure. Each participant read a short sequence of interactive instructions (Figures 1 and 2). They were told that with each round of the game they could input their best guess of the value of a secret random integer number that was within the interval [1, 1000], told that they would receive feedback indicating whether their guess was too high or too low, instructed in how to enter their guesses, and encouraged to respond efficiently and without any by-hand computation.

Following the end of the 'instructions' session, participants played the game until they guessed the secret number correctly at which point the game ended. Non-numeric and non-integer responses were not allowed. Participants were allowed to enter integers outside of the number interval of the game but those were rare and truncated to the nearest interval endpoint in post-processing. The history of participants' guesses and feedback were not visible to them as they played because we wanted to investigate whether memory load would have an effect on participants' decisions in the larger interval game.

After guessing the secret number, participants answered a Yes/No follow-up question that asked whether the participant had been familiar with the concept of binary or bisection search prior to playing this game. This was asked in case we would want to investigate an effect of this prior knowledge on strategy in future work.

Our main interest was to see how participants' guesses at every turn diverged from the EIG maximizing guess.

Results

First guesses. The first guess has no previous data to rely on. The EIG maximizing guess is to pick the midpoint of the interval which is 500. This is also what a pure bisection search algorithm would pick for the first guess. We examined whether participants select the optimal first guess.

Figure 6 shows the distribution of first guesses of the 43 participants that played the mid-sized interval version of "Guess the Number". About half of the participants (48%) selected the optimal first guess of 500.

Even though persons do not choose the exact midpoint of 500, they could have chosen a value very close to this midpoint perhaps due to noisy or heuristics computation of *floor((*100+1*)/2)* by the time-constrained human mind. We investigated this by performing a chi-square goodness-of-fit test to compare the distribution of first guesses to a Gaussian distribution centered at 500 and with standard deviation estimated from the data. The test rejected the null hypothesis that the first guess of participant's comes from such a distribution at the 5% significance level (p=1.56e-10).

Persons were told that the secret number was randomly chosen. They were not told that it was in fact *uniformly* randomly chosen from the game interval. The discrete uniform random distribution is arguably the most intuitive when one is told that a number is randomly chosen from an interval. But perhaps people have other internal priors about the distribution of random numbers? We performed another chi-squared goodness-of-fit test to test the null hypothesis that participants' first guesses were distributed according to the uniform distribution over the game interval [1, 1000]. This test failed to reject the null hypothesis at the 5% significance level (p=0.1420). This suggests that on average people believe that the secret number could be equally likely anywhere on the given number line. People have the correct internal prior about the randomness of the secret number (i.e uniform over interval). However, perhaps because of this unbiasedness some fail to make the EIG maximizing first guess which is the midpoint of 500.

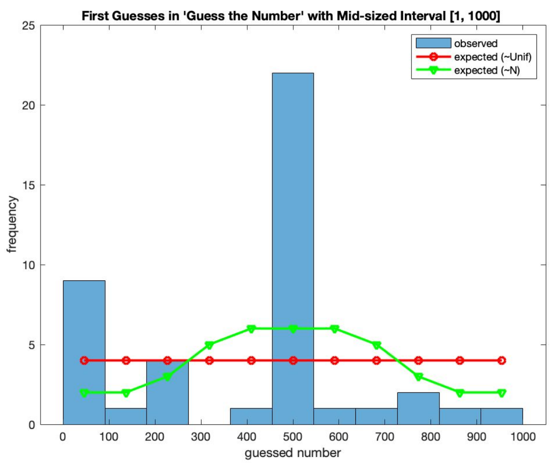


Figure 6: Histogram (blue bars) of first guesses by the 43 participants who played the mid-sized interval "Guess the Number Game". The expected counts if first guesses were distributed according to *N(*500, *σ)* and *Unif([*1, 1000*])* are also plotted in green and red, respectively (*σ* is the sample standard deviation of the first guesses).

Guess sequences. Having analyzed the first guess data, we wanted to investigate the optimality of the full sequence of guesses that participants made (i.e the guess made at every turn until guessing the secret number correctly and ending the game).

We first compared the number of guesses taken by participants to find the secret number for their games versus a pure bisection search algorithm set to the exact same task (Figure 7). For each participant, the bisection search algorithm is used to find the same secret number in the same interval and given the same feedback after each guess.

Figure 7 shows that people on average took more turns before guessing the secret number (~12) than the bisection search algorithm did (~9). The bisection search procedure on this interval has known worst-case performance of $\log_2(1000)$ ~= 10 guesses, which is why the boxplot for the algorithm in Figure 6 has no outliers greater than the upper notch. Humans, however, can be extremely unstrategic in their guesses and so large outliers are possible. Conversely, just as humans can get "lucky" and guess the correct secret number by chance early on (for example, within 2 guesses), the algorithm can get "lucky" when the secret number happens to an early bisection midpoint (like 500, 250 or 750). Therefore, outliers greater than the lower notch are possible for both human and algorithm. The variance between the number of guesses taken by people and the algorithm is also striking (34.4 human, ~2.0 algorithm). A one-way ANOVA showed that the number of guesses until game end differed significantly between humans and algorithm [F(1,43)=14.17, p=0.0003].

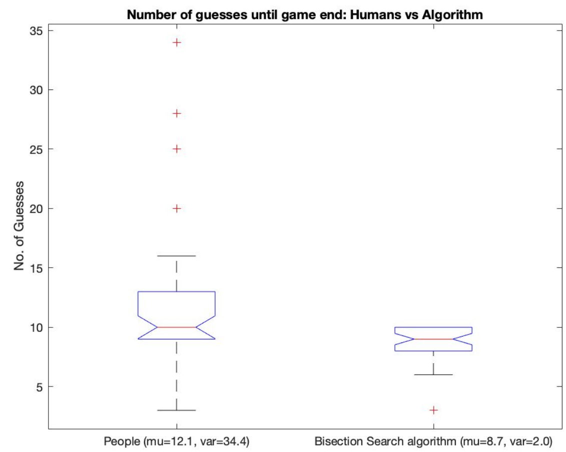


Figure 7: Humans on average take more tries (12.1) to guess the secret integer correctly than the bisection search algorithm (8.7). Humans also show more variability in the number of guesses (min=3, max=28) until game end than the algorithm (min=3, max=10). There is a significant difference between the number of guesses between the two groups (p = 0.0003).

Humans may not use pure bisection search but they should still make rational guesses conditioned on feedback received on previous guesses. We wanted to compare participants' guesses to a one-step EIG maximizing agent. To do this we parsed each person's guess sequence for the game through an algorithm that predicts the EIG maximizing guess for that participant at each turn conditioned on their previous guesses. As shown in the 'Model' section, the EIG maximizing guess bisects the smallest interval over which the posterior distribution of the secret number is non-zero. We calculated the absolute error between what participants actually guessed a given turn and what an EIG maximizing agent would guess using their previous guesses. This error was then normalized by the size of the game (1000 in this mid-sized interval version of the game"). Therefore, the maximum difference or delta possible on any turn is 1 and the minimum is 0 (person guesses the EIG maximizing value). Figure 8 plots the sequence of deltas between participants and their corresponding EIG agent over time. The posterior median of this delta was calculated at each timepoint (guess number). A 1-degree polynomial model (R^2=0.22) and a power series model (R^2=0.41) were fitted to the median errors. The coefficients of the power series model $a * x^b + c$ (with 95% confidence bounds) were:

$a =$ 0.06871 (0.03058, 0.1068)

$b =$ -0.4828 (-1.216, 0.2508)

$c =$ -0.01372 (-0.05725, 0.02981)

These results suggest that participants behave more like EIG maximizing agents as the game progresses. Therefore, they are using the information from previous guesses to make future guesses, just not in the most efficiently possible way.

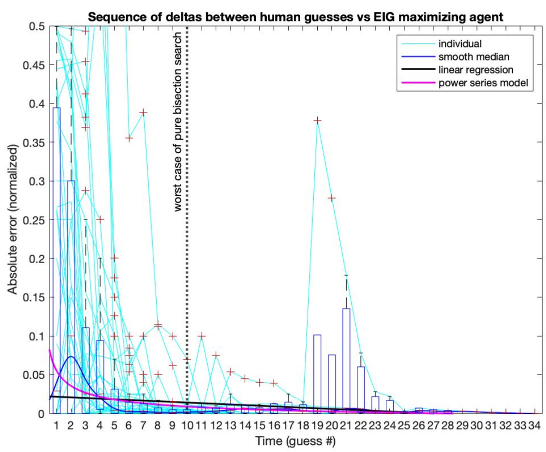


Figure 8: The divergence between humans and an EIG maximizing agent diminishes over with more guesses. Light blue lines trace the errors of individual participants. Boxplots are shown at each timepoint (i.e guess number in the game) with outliers marked by red crosshairs. A smoothing spline is fitted to the median errors (dark blue line). A 1-degree polynomial linear regression model (black line) and a power series model (purple curve) are fitted to the posterior median error.

Experiment 2: Large Interval

Method

Participants. 23 participants completed the experiment hosted online with Pavlovia.

Materials. We used the interval [1, 100000]. The secret number h* was chosen uniformly at random at the start of the experiment for each participant.

Procedure. The procedure was identical to Experiment 1 except that the upper bound of the integer number interval was now 10^5 instead of 10^3 .

Results

First guesses. The EIG maximizing guess is to pick the midpoint of the interval which is 50000. We examined whether participants select the optimal first guess. We perform an analysis similar to the corresponding section in Experiment 1.

Figure 9 shows the distribution of first guesses of the 23 participants that played the large interval version of "Guess the Number". In this version, 10 of the 23 participants (43%) selected the optimal first guess of 50000.

We again performed two chi-squared tests similar to those done for the corresponding analysis Experiment 1. This time we compared the distribution of first guesses to a Gaussian distribution centered at 50000 (standard deviation estimated from the data) and a Uniform distribution over the integer set [1 10000]. Both tests failed to reject the null at the 5% significance level (p=0.6190 for $N(50000, \sigma)$; p=0.6887 for *Unif([*1, 100000*]))*. The lack of replication of the result from Experiment 1 could be because of lack of statistical power due to fewer participants for the large interval game.

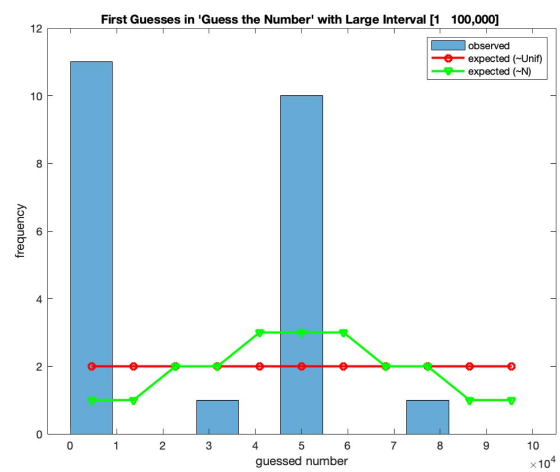


Figure 9: Histogram (blue bars) of first guesses by the 23 participants who played the large interval "Guess the Number Game". The expected counts if first guesses were distributed according to $N(50000, \sigma)$ and *Unif([*1, 100000*])* are also plotted in green and red, respectively (σ is the sample standard deviation of the first guesses).

Guess sequences. Similar analysis to those described in Experiment 1 were performed.

Figure 10 compares the number of tries taken by participants versus the optimal bisection search algorithm until the secret integer is guessed. Humans take on average 29 guesses with very large variance (~171) whereas the pure bisection search algorithm takes about 16 guesses on average with. One-way ANOVA (F(1,23)=22.54, p=2.21e-05). Therefore the number of guesses is significantly different across humans and the bisection search algorithm.

We once again computed the delta between each participants' guesses and a one one-step EIG maximizing agent conditioning on previous guesses. Figure 11 plots this sequence of deltas for the large interval version of the game. The posterior median of the errors at every guess/turn were computed and fitted with a linear regression model (R^2=0.16) and a power series model (R^2=0.69). In this version, the power series model was a better fit of the median errors than in the mid-sized interval version ($R^2 = 0.69 > 0.41$). The coefficients of the power series model $a * x^b + c$ (with 95% confidence bounds) were:

a = 0.4002 (0.3358, 0.4646)
b = -0.8248 (-1.079, -0.571)
c = -0.0226 (-0.04549, 0.000303)

The larger y-intercept of the models in this version than in Experiment 1 suggests that with a larger interval people are less likely to choose the EIG maximizing guess on their first turn. This supports our idea that people's (correct) internal prior about the randomness of secret hidden could actually contribute to their failure to always pick the midpoint as their first guess.

These results once again suggest that participants behave more like EIG maximizing agents as the game progresses. The human inability to act exactly like an EIG maximizing agent earlier on contributes to the on average greater number of guesses to find the secret integer.

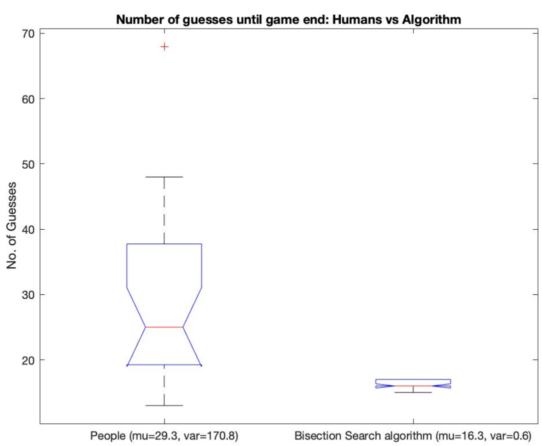


Figure 10: There is a significant difference between the average number of guesses taken between humans and the bisection search algorithm on the large interval "Guess the Number" game (p = 2.21e-05). Humans show much much larger variability (min=13, max=68) in the number of guesses taken to find the secret integer than the algorithm (min=15, max=17).

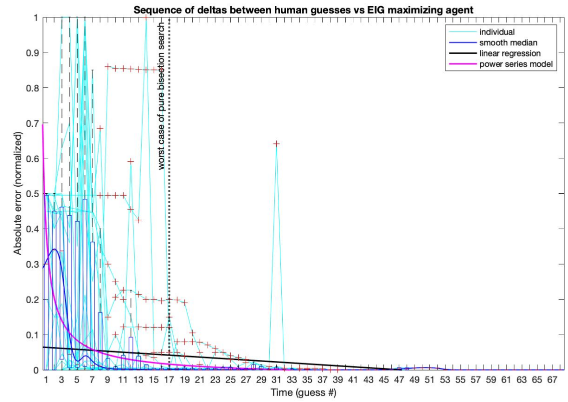


Figure 11: Participants behave more like EIG maximizing with more guesses. Light blue lines trace the errors of individual participants. Boxplots shown at each guess with outliers (red crosshairs). Smoothing spline fitted to the posterior median errors (dark blue curve). A linear regression model (black line) and a power series model (purple curve) are fitted to the posterior median errors.

Discussion

Humans are surprisingly good at solving a wide range of problems efficiently. The ability to search for information in large and noisy environments is important for our learning and the ability to use that information effectively allows us to reap rewards and avoid exploitation by more efficient actors. Computer scientists have devised algorithms for solving large classes of problems in a way that is either fast, uses little memory or computational resources, or both. These algorithms are clever in how they extract some underlying deterministic structure in the problem. The cleverness of these algorithms often clouds the very natural intuitions behind them. The world is often noisy and there is a large space of actions to choose from at every moment. Therefore, an underlying structure is often difficult to exploit in a probabilistic world. Nevertheless, humans are remarkably good at employing heuristics or basic intuitive algorithms to make decisions. Oftentimes, we make a sequence of small decisions that are not necessarily optimal but are efficient. This sequence of suboptimal local decisions can often get us close to our goal or desired state.

We wanted to investigate the optimality of people's intuitive algorithms. To do this we had to devise a problem for which a well studied and proven optimal algorithm for solving it was known. We also wanted this problem to be simple enough to implement and test. One such problem is that of finding an item in a sorted list. Binary or bisection search is an efficient algorithm for finding an item from a sorted list of items. It works by repeatedly dividing in half the portion of the list that could contain the item, until you've narrowed down the possible locations to just one (Wikipedia, 2020). By making this sorted list an integer number interval, we devised the task of guessing a secret number located in this interval. The noisiness of the real world was captured in this task by the randomness of the location of the secret number to be guessed. The large action space was modeled by the large size of the number interval. Bisection search is the most efficient algorithm to solve this and it is easy to implement a solution using code.

However, because human minds are not perfect computing machines, we hypothesized that human decision-making on this task would diverge from the optimal algorithmic procedure. We wanted to understand whether there was any structure to this divergence and possibly why it occurred. We found that indeed there is some structure to the difference between how humans solve this problem and how the optimal algorithm solves it. Specifically we found that whereas the bisection search algorithm maximizes expected information gain (EIG) from the very start, humans are initially non-strategic in their guesses but behave more like EIG maximising agents with more guesses. Our result adds to previous work on cognitive information search (Nelson et al., 2013, Tsividis wt al. 2014) and rational approximations (Sanborn et. al, 2010, Zhi-Xuan et al.). Our results suggest that human decision-making can be improved by somehow encouraging or coercing EIG maximization earlier on in a particular process.

Future work is necessary to figure out the exact way to do this but one interesting insight from our work may help. We found that on average, people's priors about the randomness of the secret number were correct since people selected initial guesses from the full range of the number interval despite not being told explicitly that the secret integer was *uniformly* randomly chosen (they were told that it was randomly chosen over the interval; however, many different probability distributions could be defined over a finite interval). Therefore, having the correct prior may actually have thrown people off of a more efficient guess early on. For example, if people had an internal prior that the secret number was drawn from a truncated Gaussian centered at the midpoint of the interval, they would be more likely to pick a more efficient (in terms of smaller distance from EIG max guess) first guess. More work is needed to investigate this phenomenon.

In our experiments, we collected feedback from all participants on whether they were familiar with bisection/binary search prior to playing the "Guess the Number". This binary classification of participants was not used in the analysis in this paper due to the lack of statistical power that would result from splitting an already small number of participants overall. However, in future work we would like to investigate the effect of prior knowledge of an algorithm on human decision-making. This is an interesting question because one might argue that the purpose of a formal education is to make us more efficient at solving problems and less likely to be exploited by more rational agents. However, the transference of an algorithm to a real-work problem is not always obvious. We may also get stuck in using our "intuitive" but often non-optimal procedures because they are more familiar to us and demand less of a cognitive load than a computer science algorithm for example.

References

Evans, Owain. "Learning the Preferences of Bounded Agents." (2015).

Griffiths, Thomas L, and Joshua B Tenenbaum. "Optimal predictions in everyday cognition." *Psychological science* vol. 17,9 (2006): 767-73. doi:10.1111/j.1467-9280.2006.01780.

Jokar, Elham, and Mohammad Mikaili. "Assessment of human random number generation for biometric verification." *Journal of medical signals and sensors* vol. 2,2 (2012): 82-7.

Sanborn, A. et al. "Rational approximations to rational models: alternative algorithms for category learning." *Psychological review* 117 4 (2010): 1144-67 .

Tsividis, P., Gershman, S., Tenenbaum, J., & Schulz, L. (2014). Information Selection in Noisy Environments with Large Action Spaces. *Proceedings of the Annual Meeting of the Cognitive Science Society*, 36. Retrieved from https://escholarship.org/uc/item/01s3b3hc

Vul, Edward et al. "One and done? Optimal decisions from very few samples." *Cognitive science* vol. 38,4 (2014): 599-637. doi:10.1111/cogs.12101

Waeber, Rolf et al. "Bisection Search with Noisy Responses." *SIAM J. Control. Optim.* 51 (2013): 2261-2279

Contribution Statement

This project began in mid-November. Quilee and I decided on the idea for the paper around then. We were fascinated by the reasons for suboptimal decisions. It makes sense that people make mistakes. But even more fascinating was the general accuracy of the global decision even when inaccurate local decisions were made enroute to that global decision. In other words, humans generally get the big decisions right. The smaller decisions that lead up to that big decision may or may not be accurate. In many cases, it doesn't seem to matter. We were initially drawn to this question.

We tried to split the project in half such that we wouldn't be blocking each other. I would build the game and collect the data. Quilee would do the analysis and discussion. I set out in mid-November to build "Guess-the-Number" using psychopy. There was a bit of a learning curve but I'm glad I did learn it. I will definitely be using it for future projects. Uploading to Pavlovia proved to be a bigger challenge. Nevertheless, I was in charge of building the game, distributing, and recruiting participants to play the game.

Once results started filtering in, Quilee began analysis. Participant-data collection continued throughout the first half of December. Quilee was finishing up analysis and took the lead on the actual writing of the paper.

I am proud of the work Quilee and I did and am excited to potentially continue this work in the future.

Ian Miller took this class for undergraduate credit.

Quilee Simeon took this class for graduate credit.