Modelling value-based decision-making policies using Genetic Programming: A proof of concept study

Angelo Pirrone
Centre for Philosophy of Natural and Social Science
London School of Economics and Political Science
London, UK

Fernand Gobet
Centre for Philosophy of Natural and Social Science
London School of Economics and Political Science
London, UK

Corresponding authors {a.pirrone, f.gobet}@lse.ac.uk. The authors declare that there is no conflict of interest regarding the publication of this article. The MATLAB code used for the simulations presented in this study is available at https://osf.io/5ma6c/. We thank Laura Bartlett and Noman Javed for comments on the manuscript. Funding from the European Research Council (ERC-ADG-835002—GEMS) is gratefully acknowledged.

Abstract

An important way to develop models in psychology and cognitive science is to express them as computer programs. However, computational modelling is not an easy task. To address this issue, it has been proposed to use artificial-intelligence (AI) techniques, such as genetic programming (GP) to semi-automatically generate models. In this paper, we establish whether models used to generate data can be recovered when GP evolve models accounting for those data. As an example, we use an experiment from decision making, which addresses a central question in decision making research: to understand what strategy, or ‘policy’, agents adopt in order to make a choice. In decision-making, this often means understanding the policy that best explains the distribution of choices and/or reaction times of two-alternative forced-choice decisions. We generate data from three models using different psychologically plausible policies. We then evaluate the ability and extent of GP to correctly identify the true generating model, among the class of virtually infinite candidate models. Our results show that, regardless of the complexity of the policy, GP can correctly identify the true generating process. In view of these results, we discuss implications for cognitive science research and computational scientific discovery, and possible future applications.

Keywords: genetic programming, value-based decision-making, cognitive modelling, cognitive science

Modelling value-based decision-making policies using Genetic Programming: A proof of concept study

Introduction

Two key aspects of scientific discovery are the generation of predictions, and the development of models. In psychology and cognitive science, the generation of predictions often refers to predicting participants' observable behaviour. The generation of models instead refers to elucidating the combination of, mostly unobservable, mechanisms and/or processes that give rise to a specific behaviour.

An important means to develop models in psychology and cognitive science is to express them as computer programs. Such models offer the advantages of being unambiguous, explaining both simple and complex behaviour, and making clear-cut predictions (e.g. Gobet et al., 2011). However, computational modelling is not an easy task. At the very least, it requires acquiring skills in computer science and programming in addition to skills specific to a particular domain, such as psychology. In addition, the generation of scientific models can be described as a heuristic search in the combinatorial space of all the possible candidate models that explain a specific phenomenon (Frias-Martinez & Gobet, 2007; Simon, 1977). Given the infinite size of such spaces, searching them can be very hard indeed both theoretically and computationally, and human scientists can explore only a limited portion of those spaces. One way to alleviate these difficulties is to use artificial-intelligence (AI) techniques to (semi-)automatically develop models. In particular, AI has developed a number of search techniques that can semi-automatically perform an efficient search in these spaces.

The aim of this article is to show how a specific search technique, genetic programming (GP; Koza, 1992), can be used to support the generation of models in cognitive science. Genetic programming evolves a large number of computer programs applying principles based on natural evolution, using as fitness value the extent to which the programs solve target problems. Our approach will be to generate synthetic data from known models and evaluate whether GP can correctly recover the models that generated the data. As a domain of study, we use a well-known experiment from research into value-based decision-making, and select three simple, yet psychologically plausible models that guide decision-making. We focus on establishing whether GP can discover back the decision making policies (strategies) that were implemented in the models.

Value-based decision making

In value-based decision-making (for examples, see Tajima et al., 2019; Tajima et al., 2016), tasks consist of comparing the values of rewarding alternatives. Classical examples are foraging scenarios, and consumer choices. Compared to perceptual decision-making (Bogacz et al., 2006), in which participants make a decision mostly on the basis of sensory evidence (e.g., decide whether a noisy visual stimulus is tilted clockwise or anticlockwise, or decide which of two stimuli is brighter), in value-based decision-making (Krajbich et al., 2010; Krajbich et al., 2012) choices are also affected by the expected utility associated with alternatives. Research has shown that the policy that agents use in value-based decision-making is affected by a number of factors, such as the number of alternatives (Churchland & Ditterich, 2012) or the visual fixation patterns (Krajbich et al., 2010).

Genetic Programming

GP evolves a population of candidate models in the form of computer programs in order to minimise an objective fitness function (in our case, the difference between the model's predictions and the human data). From one generation to the next, evolutionary mechanisms such as mutation and crossover allow the candidate models to evolve and outperform the previous generation in minimising the fitness function. In GP, models are generated by combining *terminals*, the inputs given to the models, and *operators*, the operations that GP can perform on the terminals. Both terminals and operators are defined by the researcher.

The model space is a function of the number of terminals and operators – but also of additional parameters such as limits on the complexity of the tree, or the sampling method (see Koza, 1992; Silva and Almeida, 2003). GP constructs *trees* that represent the relationship between operators and terminals. Figure 1 shows an example of a GP tree; this tree was estimated by providing as input four arbitrary values X1, X2, X3 and X4 and as operators the ability to subtract (the 'minus' operator) and multiply (the 'times' operator) those inputs. The tree of Figure 1 only reads inputs X1, X2 and X4 (i.e., it does not include X3 in its solution). The value on the top-right of the tree, X4, is multiplied by the difference between two further operations; the operation on the sub-tree on the bottom-left multiplies X1 and X4, while the operation on the bottom-right of the tree subtracts X2 from X2, resulting in 0 (that is, the value of X2 is irrelevant). Hence the tree of Figure 1 reduces to multiplying X4 by the product of X1 and X4.

In GP, *mutation* allows random changes in the tree structure, for example by substituting the 'times' on top of Figure 1 with a 'minus'. By contrast, *crossover* selects a random sub-tree (i.e., a section of the tree) from two different trees and swaps them. Selection governs the probability that a tree is replicated in the next generation; a common selection mechanism is that a tree will be replicated in the next generation proportionally to its fitness (in our case, the variance it explains compared to other trees). Other mechanisms such as *shrink mutation* or *swap mutation* are available; however, in our current work we exclusively use mutation and crossover.

GP has a long history and has had many applications, including antenna designs (Lohn et al., 2004), patented electronic circuits (Koza et al., 2004) and molecular structure optimisation in chemistry (Deaven & Ho, 1995). In cognitive science there have been some applications of GP to improve curve fitting (Hollis et al., 2006), to discover variable interactions (Westbury et al., 2003), and to evolve models (Frias-Martinez & Gobet, 2007; Gobet & Parker, 2005).

There are a number of benefits for estimating solutions to a problem using GP; first, GP naturally overcomes local minima problems and sensitivity to the values of initial parameters that affect other minimisation procedures (Frias-Martinez & Gobet, 2007; Koza, 1992). Moreover, GP allows the investigation of large portions of the space of possible models given a number of operators, as opposed to testing a single model that the researcher wants to verify or falsify. This reduces the risk of confirmation bias which often drives hypothesis testing in cognitive science (Bilalić et al., 2010).

Methods and simulations

Consider a scenario in which agents are presented with two alternatives. In a classical experimental setting (e.g., Pirrone et al., 2018), this usually means that alternatives are presented at the same distance from a fixation point to the left and to the right of a computer screen. In a real-life setting, such as consumer choices, this would mean that alternatives are presented at the same distance from the initial position of the agent.

Agents have learnt, over the course of previous similar encounters with such alternatives, that the values of alternatives vary in the arbitrary range 1-10, with the worst option having a value of 1, and the best option having a value of 10. During each trial (i.e., each encounter with two alternatives), agents need to choose one of the alternatives and are rewarded on the basis of the value of the alternative chosen, regardless of whether the alternative chosen is the best. In particular, we assume that the two alternatives are presented for one second and after the presentation time, agents are prompted to choose one of the two.

Agents use different strategies to choose between the two values. We are interested in whether, based on the choices made by an agent, GP can identify the strategy used to generate the data. For each trial we randomly selected the value of the left and right alternative, from a discrete distribution of possible values. We simulated a total of 1,000 trials and for each applied the agent's strategy to make a choice.

Genetic Programming: Implementation and operators

For the sake of brevity, given our ‘proof of concept’ focus, we will provide minimal reference to the core and most important aspects of GP for our application; readers interested in more details about GP should refer to exhaustive books and tutorials (Banzhaf et al., 1998; Koza, 1992; Langdon & Poli, 2013; Poli et al., 2008). We used GPLAB, an excellent and versatile MATLAB toolbox (Silva & Almeida, 2003), to run GP using as terminals the values of the two alternatives, a random integer number generator in the range 1-10, which captures the range of values of the alternatives, and a random number generator between zero and one. X1 is the value of the alternative on the left and X2 the value of the alternative on the right, while numerical values from the GPLAB’s random number generator are reported in the estimated trees.

We adopted the following operators: ‘gt’ (i.e., greater) – this operator computes if element A is greater than element B and it outputs 0 or 1 depending on whether the result of the comparison is false or true; ‘le’ (i.e., less) – this operator computes if element A is lower than element B and it outputs 0 or 1 depending on whether the result of the comparison is false or true; ‘plus’ – this operator sums two elements; ‘minus’ – this operator subtracts the value of two elements; ‘times’ – this operator multiplies the value of two elements; and ‘mydivide’ – this operator divides the value of two elements. If the value of the divisor is equal to zero, ‘mydivide’ outputs the value of the dividend; that is, if B = 0, then mydivide(A,B) = A.

Given the simplicity of our simulated scenarios and of the type of policies that participants can adopt in these scenarios, in order to avoid ‘bloating’ (the tendency in GP for programs to grow very large) and overfitting, we imposed a strict limit of five nodes to solutions estimated using GP. We set default values for all other GP parameters; these can be accessed in Table 3.2 of GPLAB’s manual (Silva & Almeida, 2003). In order to minimise the discrepancy between actual choices and GP predicted choices, we ran GP with 500 individuals (i.e., 500 models) that were allowed to evolve for 500 generations. The simulations were run with three different models that respectively used the following strategies: a 'satisficing' policy, a relative policy, and a relative policy with bias. In a second set of simulations, we removed the limit concerning the maximum number of nodes allowed for each individual.

First scenario: A 'satisficing' policy

One policy that may apply to value-based choices is a so-called 'satisficing' policy: if the value of one of the alternatives is higher than a threshold of acceptability, choose that alternative, otherwise choose the other alternative. Note that regardless of specific simplifications and assumptions that we are making, this is a simple policy for value-based decision-making that is known in economics (Simon, 1959) and behavioural ecology (Kacelnik et al., 2011; Pirrone et al., 2014) and that makes it possible to break decision deadlock over difficult discriminations, in the presence of time costs associated with longer decisions.

Let $P_{v_{left_i}}$ be the probability of choosing left for trial i. In this scenario, agents only focus on the alternative presented on the left, v_{left_i} and, after one second of presentation, they decide whether the value of the alternative plus that of arbitrary white Gaussian noise with variance .01 randomly sampled once every millisecond is higher than a threshold of acceptability, defined as the mean value of the possible range of alternatives μ_v(that is equal to 5.5 in our example). It is important to add noise in the decision making process for two reasons. First, it makes it possible to account for known across-trials variability in choices. That is, if the same trial is repeated multiple times, agents' choice may vary. Second, if value representations are not noisy, it is not clear why a comparison process should even take place, as agents would be expected to make almost instantaneously a choice in favour of the most valuable alternative, which is not the case in value-based decision-making, nor in perceptual decision making (Bogacz et al., 2006; Krajbich et al., 2010).

Put formally, for each trial the simulated choice probability (defined as the probability to choose left) is given by

$$P_{v_{left_i}} = v_{left_i} + \epsilon_i > \mu_v \tag{1}$$

The best model estimated from GPLAB is reported in Figure 2A. This simple tree estimates whether X1 (the alternative on the left) is greater than 5. Note that the rule estimated by the tree is the one that we have used to generate the data; that is, agents choose the left alternative if this is higher than five, otherwise they choose right. Thus, even in the presence of non-modelled noise, GP can accurately estimate the true process that has generated the data.

Figure 3 shows a psychometric function for each of the three experimental scenarios with the probability of choosing left as a function of the difference in value between the item on the left and the item on the right (left panel), and as a function of the item on the left only (right panel). The data are displayed in black, and the predictions of the best tree estimated by GP are displayed in red. It is important to visually inspect the goodness of fit in order to understand mismatches between data and model predictions.

As expected, the simple policy of Figure 2A predicts choices well (it explains 95% of the variance of the data). The top-right panel of Figure 3 shows mean choice as a function of the left rating only. *Prima facie*, this panel might be interpreted as a mismatch between the model and the data when the value of the left rating is five. However, in the absence of noise, the trend predicted by the true generating process is the one estimated by GP. If anything, the fact that the best model does not overfit the noise is a reassurance of GP's ability to estimate the simple policy for value-based decision-making that generated the data.

Second scenario: Relative policy

In the second scenario, agents do not compare one alternative to a threshold of acceptability, as in the first scenario, but compute the difference in evidence between the value of the left and the value on the right; when prompted to make a decision, they choose left if the difference between the two items is positive, and right otherwise.

Hence, this time the choice policy (i.e., the probability to choose left) is

$$P_{v_{left_i}} = v_{left_i} - v_{right_i} + \epsilon_i > 0 \quad (2)$$

Note that this simple rule represents the core of celebrated and widely popular drift diffusion models (Ratcliff & McKoon, 2008; Ratcliff et al., 2016) of decision making, which have been applied to a large number of tasks in decision making and are statistically optimal for managing speed-accuracy trade-offs for decisions with stationary distributions of evidence (Bogacz et al., 2006) and for value-based decision-making under specific constraints (Tajima et al., 2016).

Figure 2B shows the best tree evolved by GP. Again, GP can correctly estimate the true policy that has generated the data, since the tree computes whether the left item value is higher than the right item. That is, if the value of the left alternative is higher than the value of the right alternative, the left alternative is selected, otherwise the right alternative is selected. The middle panel of Figure 3 shows good agreement between the data and the model's predictions; in this case, the model accounts for about 95% of the data.

Third scenario: Relative policy with bias

The third policy is a simple mathematical modification of the second policy. We assume that agents assign different weights to the value of the alternatives; in particular, agents discount the value of the right alternative by a factor of .3. Put formally, the probability to choose left for a trial is determined by

$$P_{v_{left_i}} = v_{left_i} - .3 \cdot v_{right_i} + \epsilon_i > 0 \quad (3)$$

This policy is qualitatively similar to that of attentional drift diffusion models (Krajbich et al., 2010), in which visual fixations play a key role in the decision-making process. In particular, in this model, the value of the non-fixated item is discounted by a factor of about .3 (Krajbich et al., 2010; Smith & Krajbich, 2018), giving rise to a number of interesting fixation-dependent biases in decision making, such as last fixation biases. Hence, we again are simulating a psychologically plausible policy for making a decision.

Figure 2C shows that GP estimates the policy of Equation 3 correctly, since the tree can be simplified to the policy of Equation 3. The tree of Figure 2C computes whether X1 is greater than X2 divided 4; this is equivalent to the rule that we have simulated (however, note that the discount factor is estimated as 0.25, while in Equation 3 it is 0.3). The bottom panel of Figure 2C shows good agreement between the data and the model, which accounts for about 97% of the variance.

Qualitatively equivalent solutions and overfitting

In the previous run, we imposed on GP a strict limit regarding the maximum number of nodes allowed to avoid bloating and overfitting of noise. We now show the results from a second run in which we removed this strict limit on GP solutions. This second run allows us to show (a) the ability of GP to estimate *different*, qualitatively similar, solutions to a specific problem, and (b) the importance of avoiding overfitting of results.

Figure 4 shows the trees estimated by a second run of GP without a strict limit, while Figure 5 shows the comparison between the data and the model. For the first scenario, Figure 4A, the policy estimated is similar to the true generating process, in which participants choose the left alternative if it is higher than a threshold of acceptability – although in this case the threshold of acceptability is estimated as four, rather than five.

For the second scenario, the policy in Figure 4B attempts to overfit the noise as shown in the middle panel of Figure 5; this is achieved by adding an unreasonable number of operations to the best solution in order to account, for example, for the fact that when the difference between left and right alternative is zero, choice is at chance level. However, a simulation of this policy over all the combinations of possible left and right alternatives (excluding cases for which the left and right alternatives are identical) has shown that this policy is qualitatively equivalent to the data generating process, given that it always chooses the alternative with a higher value. However, the complexity of the tree does not provide a direct insight into the policy that generated the data.

For the third scenario, the policy in Figure 4C computes whether the ratio between the discounted value on the right and the value on the left is bigger than four. In this case, this tree is equivalent to computing whether the difference between the value on the left and the discounted value on the right is higher than zero. Mathematically, this rule is undistinguishable from the rule that we used to generate the model.

Discussion

We simulated three different decision making policies and estimated the ability of GP to correctly find the known data-generating processes. In all cases in which we set a strict limit to GP, such policies were successfully recovered.

Given the high stochasticity of GP, different runs can give rise to different solutions. This means that GP will find *different* solutions over different runs. However, in the case of non-highly dimensional problems, such as policies for simple two-alternative forced-choice decisions, different trees will often represent qualitatively similar solutions. Also, since we decided not to model the noise in the data, GP estimates will vary across runs because of the different ways in which the noise affects the data or the tree represents the variance that is due to the noise. For example, over two different runs, we have shown that the solution found by GP was qualitatively correct, but estimated a threshold of acceptability of four, rather than five, for the first scenario. This is expected given non-modelled noisy variations in the data and high stochasticity of GP. In disciplines such as cognitive science, data have multiple sources of noise, both at the individual and inter-individual level. Setting strict limits to GP results can overcome the overfitting of noisy variations, and optimise the trade-off between simplicity and goodness of fit of solutions.

It is also possible, however, in the case of high dimensional problems, that data are explained by a number of contrasting and not necessarily qualitatively similar explanations. Take the example of the second scenario; a policy in which participants choose left if the difference between items is higher than zero is undistinguishable from a policy in which participants choose the left item if the ratio between left and right values is higher than one. Also, given our simulated data, the tree estimated in Figure 2C is undistinguishable from that estimated in Figure 4C. For the researcher interested in understanding policies for decision making, choosing between equally good solutions translates into collecting more data to identify the model or, even better, conducting studies with specific experimental manipulations that can only be met by one policy. Hence, GP makes falsifiable predictions that can be used to drive future experiments in order to discriminate among alternative models. However, it is also possible to discard solutions found by GP on the basis of knowledge from previous reliable results/theories. For example, while this is not the case (Pirrone et al., 2018), assume that previous research excluded the possibility that participants compute over the course of a trial the overall magnitude of the alternatives; in this case, a GP solution that computes the overall magnitude of alternatives may be excluded based on theoretical grounds.

Often, GP trees can be difficult to interpret and need to be post-processed in order to be simplified, for example by removing redundancies. In our study, trees were simple enough to be interpreted without the need of post-processing. However, for complex trees, the result from dozens, or even hundreds, of operations can be greatly simplified (for example when the result of a number of operations is always equal to a constant) but often this cannot be understood by visual inspection of the tree. In the case of complex trees, dedicated algorithms can support and automate the post-processing of trees (Garcia-Almanza & Tsang, 2006; Rockett, 2020).

Although the simple policies that we have simulated here could have also been estimated through general linear models, which are known to most researchers in cognitive science, GP offers a number of advantages. First, GP does not require any assumptions regarding the data as opposed to general linear models (e.g., normally distributed dependent variable, normally distributed errors). This is particularly important for analyses of reaction times, a key dependent variable in cognitive science, which are generally positively skewed and for which transformations such as log-transformation to approximate a normal distribution can produce detrimental outcomes (Schramm & Rouder, 2019). Second, decision-making dynamics are often characterised by non-linearity (for value-based decision-making, see for example Pais et al., 2013); compared to linear models that by definition cannot account for those dynamics, GP can naturally be applied to model non-linear dynamics. Third, GP can provide various solutions to a problem, and as such can innovate previous accounts in cognitive science, while results from general linear models would always account for a unique solution to a specific problem. Furthermore, GP, compared to methods that are theory-driven and require an a priori formulation of candidate models from which to identify strategies (Bröder & Schiffer, 2003; Glöckner, 2009; Hilbig & Moshagen, 2014; Jekel et al., 2010; Lee, 2016), allows a 'theory-free' estimation of strategies; compared to classical black-box machine learning algorithms (Alpaydin, 2020), GP exposes the relationship between inputs in an explainable fashion.

In our simulations, decision time was exogenously triggered by a hypothetical experimenter and agents provided an answer only when asked to do so, after one second. Since information regarding decision times was uninformative, we focused on choice and not on the reaction times associated with the choice. In *free-response paradigms*, in which participants can make a decision in their own time, policies for decision making need to account simultaneously for the distribution of choices and reaction times; this will be a focus of future research using a similar approach.

For the future, we propose a body of work that will extend the simple, yet effective, rationale that we have applied here. In particular, research using GP in decision making could apply the methodology used here to data from human/animal studies in both laboratory and ecological settings. In those scenarios, the true data-generating process is unknown and a number of qualitatively similar models could have generated such models, an aspect known as model mimicry (Bose et al., 2020).

Furthermore, future research could investigate decision making policies when additional factors are taken into consideration; for example, visual fixations that can guide the computation and comparison of values (Krajbich et al., 2010), or individual differences in decision making that may be explained by different strategies for different clusters of participants. While existing models have been proposed for such factors, we believe that a more principled approach requires evaluating such models against a large number of candidate models, given a number of plausible operators.

In addition, future research will be aimed at creating and extending psychologically plausible operators, rather than simple algebraic and logical operations adopted here. For example, a psychologically plausible operator could be one that writes inputs into a visual short-term memory buffer as proposed in the literature (Frias-Martinez & Gobet, 2007; Gobet & Parker, 2005), or one that directs visual fixations to one of two items under consideration. Also, parameters such as leak in evidence accumulation or competition between alternatives (Bogacz et al., 2006), which can play a role in decision making, could be added to GP.

An interesting area of application for future research is that of decision-making with multiple alternatives (Gluth et al., 2020), that is, scenarios in which agents are presented with more than two alternatives, as is often the case in real life settings. As the number of alternatives increases, the number of policies that participants could adopt increases exponentially, and GP will undoubtedly provide useful insights for these high-dimensional complex problems.

We want to emphasise that our results do not represent a theoretical innovation *per se*. Since GP is agnostic about the nature of the data, showing that GP can account for models that generated data is expected on the basis of previous numerous application of this technique to various domains. However, here we address mainly psychologists, cognitive scientists and decision making modellers to make GP, and evolutionary computation in general, more accessible to them and thus motivate further research and applications using this technique. The assumption, explored in another line of research (Frias-Martinez & Gobet, 2007; Gobet & Parker, 2005; Lane et al., 2016), is that GP can also identify (unknown) models from real data. When it comes to develop computational models of human behaviour, GP and other forms of evolutionary computation provide a powerful means of searching through the immense space of possible models.

References

Alpaydin, E. (2020). *Introduction to machine learning*. MIT press.

Banzhaf, W., Nordin, P., Keller, R. E., & Francone, F. D. (1998). *Genetic programming*. Springer.

Bilalić, M., McLeod, P., & Gobet, F. (2010). The mechanism of the Einstellung (set) effect: A pervasive source of cognitive bias. *Current Directions in Psychological Science*, *19*(2), 111–115.

Bogacz, R., Brown, E., Moehlis, J., Holmes, P., & Cohen, J. D. (2006). The physics of optimal decision making: A formal analysis of models of performance in two-alternative forced-choice tasks. *Psychological Review*, *113*(4), 700–765.

Bose, T., Pirrone, A., Reina, A., & Marshall, J. A. (2020). Comparison of magnitude-sensitive sequential sampling models in a simulation-based study. *Journal of Mathematical Psychology*, *94*, 102298.

Bröder, A., & Schiffer, S. (2003). Bayesian strategy assessment in multi-attribute decision making. *Journal of Behavioral Decision Making*, *16*(3), 193–213.

Churchland, A. K., & Ditterich, J. (2012). New advances in understanding decisions among multiple alternatives. *Current Opinion in Neurobiology*, *22*(6), 920–926.

Deaven, D. M., & Ho, K.-M. (1995). Molecular geometry optimization with a genetic algorithm. *Physical Review Letters*, *75*(2), 288–291.

Frias-Martinez, E., & Gobet, F. (2007). Automatic generation of cognitive theories using genetic programming. *Minds and Machines*, *17*(3), 287–309.

Garcia-Almanza, A. L., & Tsang, E. P. (2006). Simplifying decision trees learned by genetic programming. *2006 IEEE International Conference on Evolutionary Computation*, 2142–2148.

Glöckner, A. (2009). Investigating intuitive and deliberate processes statistically: The multiple-measure maximum likelihood strategy classification method. *Judgment and Decision Making*, *4*(3), 186.

Gluth, S., Kern, N., Kortmann, M., & Vitali, C. L. (2020). Value-based attention but not divisive normalization influences decisions with multiple alternatives. *Nature Human Behaviour*, 1–12.

Gobet, F., Chassy, P., & Bilalić, M. (2011). *Foundations of cognitive psychology.* London, McGraw Hill.

Gobet, F., & Parker, A. (2005). Evolving structure-function mappings in cognitive neuroscience using genetic programming. *Swiss Journal of Psychology*, *64*, 231–239.

Hilbig, B. E., & Moshagen, M. (2014). Generalized outcome-based strategy classification: Comparing deterministic and probabilistic choice models. *Psychonomic Bulletin & Review*, *21*(6), 1431–1443.

Hollis, G., Westbury, C. F., & Peterson, J. B. (2006). Nuance 3.0: Using genetic programming to model variable relationships. *Behavior Research Methods*, *38*(2), 218–228.

Jekel, M., Nicklisch, A., & Glöckner, A. (2010). Implementation of the multiple-measure maximum likelihood strategy classification method in R: Addendum to Glöckner (2009) and practical guide for application. *Judgment and Decision Making*, *5*(1), 54.

Kacelnik, A., Vasconcelos, M., Monteiro, T., & Aw, J. (2011). Darwin's "tug-of-war" vs. starlings' "horse-racing": How adaptations for sequential encounters drive simultaneous choice. *Behavioral Ecology and Sociobiology*, *65*(3), 547–558.

Koza, J. R. (1992). *Genetic Programming: On the programming of computers by means of natural selection.* MIT press.

Koza, J. R., Keane, M. A., & Streeter, M. J. (2004). Routine automated synthesis of five patented analog circuits using genetic programming. *Soft Computing*, *8*(5), 318–324.

Krajbich, I., Armel, C., & Rangel, A. (2010). Visual fixations and the computation and comparison of value in simple choice. *Nature Neuroscience*, *13*(10), 1292–1298.

Krajbich, I., Lu, D., Camerer, C., & Rangel, A. (2012). The attentional drift-diffusion model extends to simple purchasing decisions. *Frontiers in Psychology*, *3*, 193.

Lane, P. C. R., Sozou, P. D., Gobet, F., & Addis, M. (2016). Analysing psychological data by evolving computational models. In A. Wilhelm & H. Kestler (Eds.), *Analysis of large and complex data* (pp. 587–597). New York, Springer.

Langdon, W. B., & Poli, R. (2013). *Foundations of genetic programming*. Springer Science & Business Media.

Lee, M. D. (2016). Bayesian outcome-based strategy classification. *Behavior Research Methods*, *48*(1), 29–41.

Lohn, J., Hornby, G., & Linden, D. (2004). Evolutionary antenna design for a NASA spacecraft. *Genetic Programming Theory and Practice II*, 301–315.

Pais, D., Hogan, P. M., Schlegel, T., Franks, N. R., Leonard, N. E., & Marshall, J. A. (2013). A mechanism for value-sensitive decision-making. *PloS One*, *8*(9).

Pirrone, A., Azab, H., Hayden, B. Y., Stafford, T., & Marshall, J. A. (2018). Evidence for the speed–value trade-off: Human and monkey decision making is magnitude sensitive. *Decision*, *5*(2), 129.

Pirrone, A., Stafford, T., & Marshall, J. A. (2014). When natural selection should optimize speed-accuracy trade-offs. *Frontiers in Neuroscience*, *8*, 73.

Poli, R., Langdon, W. B., McPhee, N. F., & Koza, J. R. (2008). *A field guide to genetic programming*. Lulu.com.

Ratcliff, R., & McKoon, G. (2008). The diffusion decision model: Theory and data for two-choice decision tasks. *Neural Computation*, *20*(4), 873–922.

Ratcliff, R., Smith, P. L., Brown, S. D., & McKoon, G. (2016). Diffusion decision model: Current issues and history. *Trends in Cognitive Sciences*, *20*(4), 260–281.

Rockett, P. (2020). Pruning of genetic programming trees using permutation tests. *Evolutionary Intelligence*.

Schramm, P., & Rouder, J. (2019). Are reaction time transformations really beneficial? *PsyArXiv*.

Silva, S., & Almeida, J. (2003). GPLAB - a genetic programming toolbox for MATLAB. *Proceedings of the Nordic MATLAB Conference*, 273–278.

Simon, H. A. (1959). Theories of decision-making in economics and behavioral science. *The American Economic Review*, *49*(3), 253–283.

Simon, H. A. (1977). *Models of discovery: And other topics in the methods of science.* Dordrecht: Reidel.

Smith, S. M., & Krajbich, I. (2018). Attention and choice across domains. *Journal of Experimental Psychology: General*, *147*(12), 1810–1826.

Tajima, S., Drugowitsch, J., Patel, N., & Pouget, A. (2019). Optimal policy for multi-alternative decisions. *Nature Neuroscience*, *22*(9), 1503–1511.

Tajima, S., Drugowitsch, J., & Pouget, A. (2016). Optimal policy for value-based decision-making. *Nature Communications*, *7*(1), 1–12.

Westbury, C., Buchanan, L., Sanderson, M., Rhemtulla, M., & Phillips, L. (2003). Using genetic programming to discover nonlinear variable interactions. *Behavior Research Methods, Instruments, & Computers*, *35*(2), 202–216.

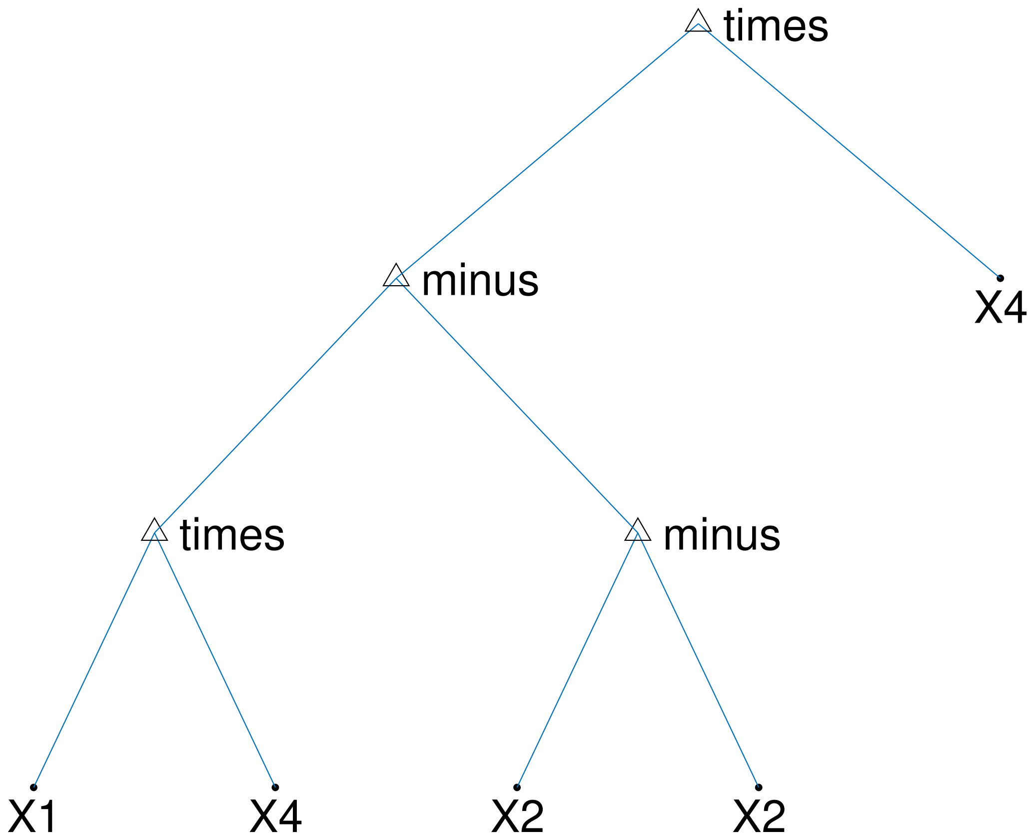

Figure 1. Example of a GP tree. This tree was generated using the GPLAB toolbox (Silva & Almeida, 2003) for MATLAB.

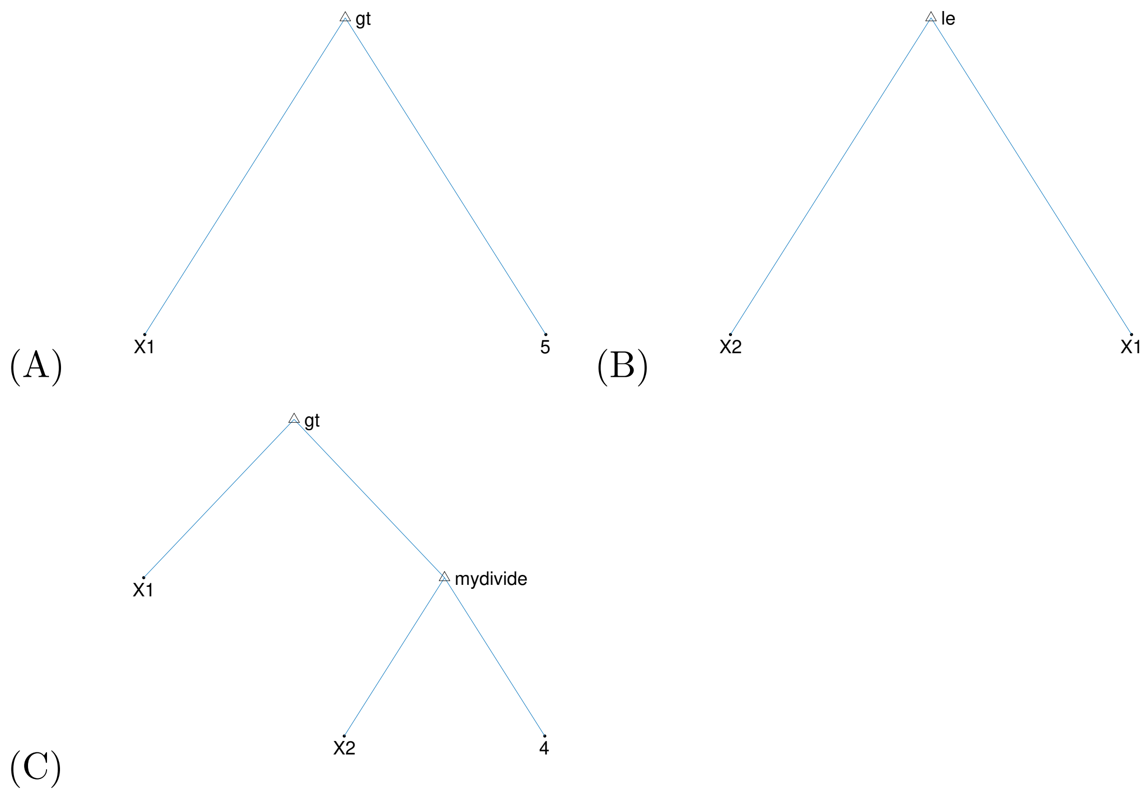

Figure 2. The best trees from the first scenario (A), second scenario (B) and third scenario (C) estimated by GP with a population of 500 individuals which evolves for 500 generations. A strict limit of five nodes was imposed.

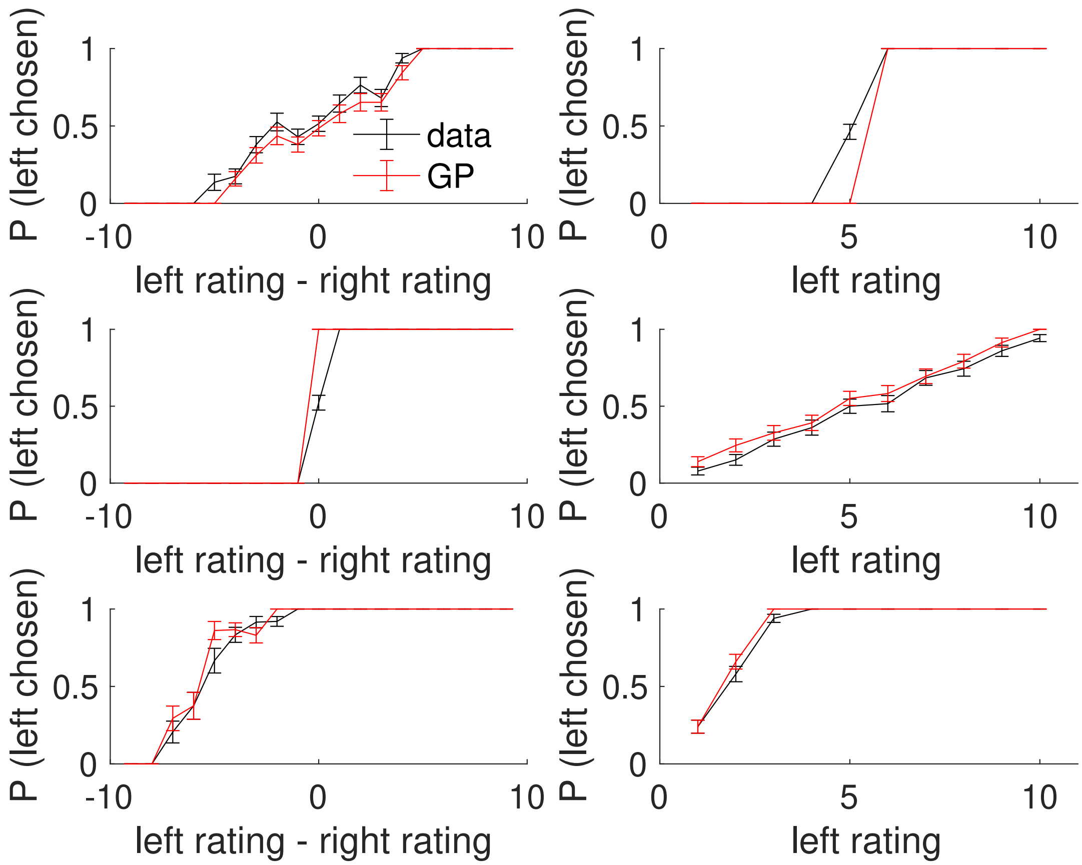

Figure 3. Comparison of mean choice (defined as the probability of choosing left) as a function of the difference in value between the two alternatives (left plots), and as a function of the value of the left alternative alone (right plots), for the first scenario (top panel), second scenario (middle panel) and third scenario (bottom panel). The data are reported in black and GP's predictions in red. Error bars are standard errors of the mean.

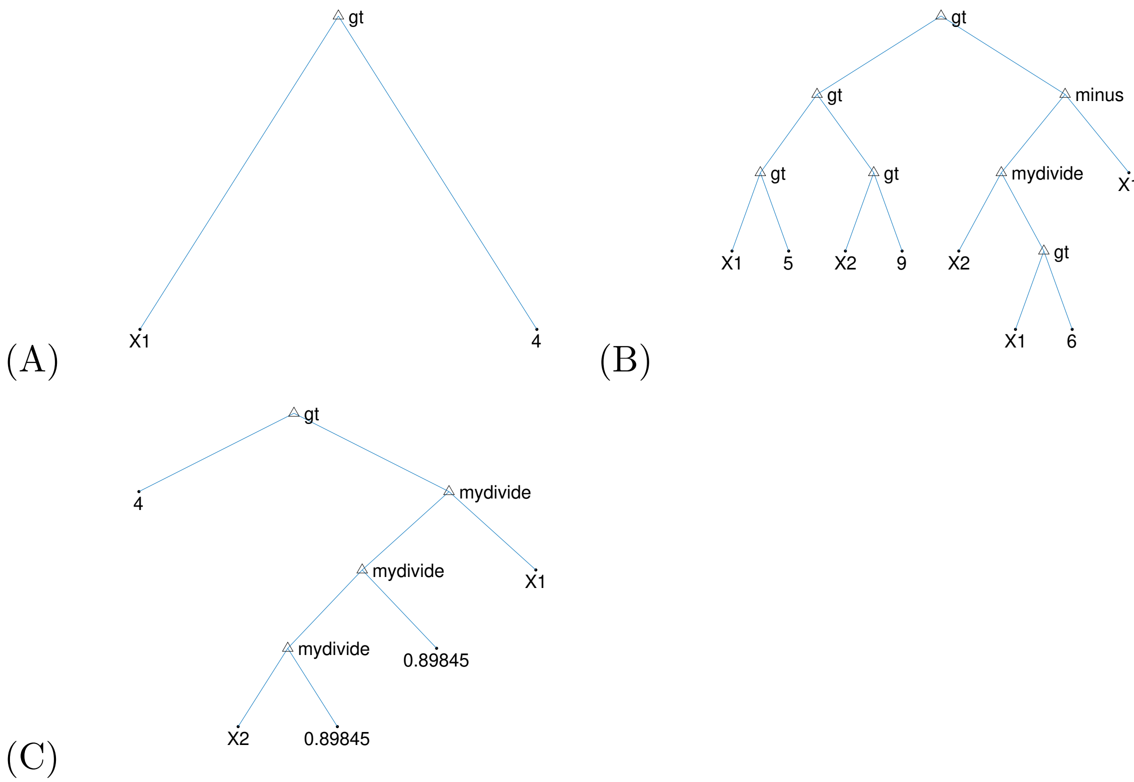

Figure 4. The best trees from the first scenario (A), second scenario (B) and third scenario (C) estimated with a second run of GP with a population of 500 individuals which evolves for 500 generations. No strict limit was imposed.

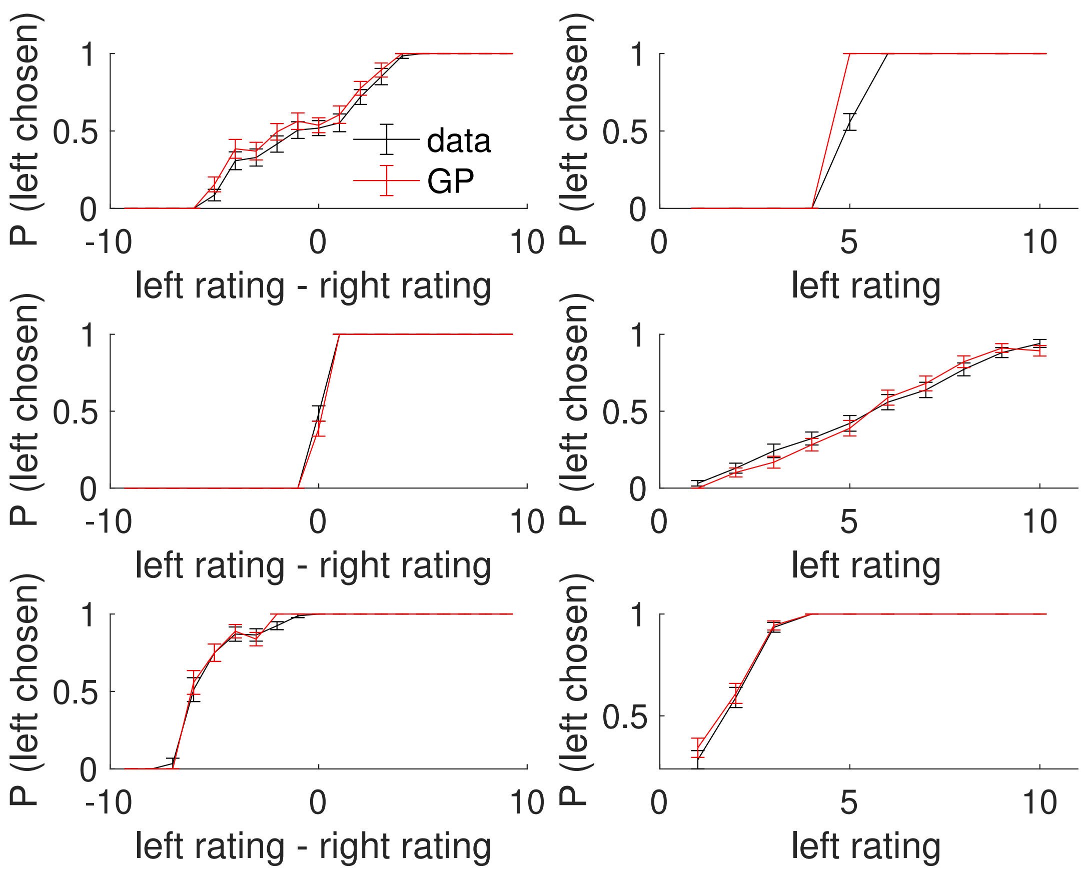

Figure 5. Comparison of mean choice (defined as the probability of choosing left) as a function of the difference in value between the two alternatives (left plots), and as a function of the value of the left alternative alone (right plots), for the first scenario (top panel), second scenario (middle panel) and third scenario (bottom panel). The data are reported in black and GP's predictions in red. Error bars are standard errors of the mean.