Computational Models of Decision Making:

The Effect of Computational Complexity on Neural Reaction Time

Peter Grabowski

Princeton University

Recent research conducted by Nathaniel Daw (Daw et al, 2011) has suggested that two key processes underlie human decision making. The first, referred to as "model free" decision making, encompasses choices that are instinctual or habitual, while the second, known as "model based" involves choices that are planned or reasoned. The current study combines Daw's findings with theories of computational complexity to make predictions about the correlation between the prevalence of model based decision processes in a subject and their average reaction time. Current theory predicts that as computational tasks become more complex, the time required to complete them increases. An analysis of the logic behind model free and model based decision making reveals that model based decisions are much more computationally complex. As such, if the laws governing computational complexity and processing time can be applied to neural circuitry in the same way they are applied to silicon circuitry, subjects who are more model based in their decision making should have slower reaction times. Our study replicates and extends Daw's finding in an attempt to determine the relationship between the relative strength of model free and model based decision making processes displayed by a subject and how quickly they responded

1. Introduction

1.1 Overview

Recent research done by Nathaniel Daw and his colleagues at New York University (Daw et al, 2011) has suggested that the processes underlying human decision making can be separated into one of two categories. The first, referred to as "model free" decision making, encompasses choices that are instinctual or habitual. The second is the converse of the first, and is referred to as either "model based" or "goal directed" behavior. It is heavily reliant on planning, and makes use of a "model," or internal representation of the environment, that functions as a sort of sandbox for testing the results of potential actions before they are taken. By using an elegantly constructed decision task, coupled with an array of algorithms borrowed from the field of dynamic programming, Daw was able to not only identify that human decision making relied on both types of process, but also to determine what percent of decision making each process's contributions were responsible for in a given subject.

The current study combines Daw's findings with theories of computational complexity to make predictions about the correlation between the prevalence of model based decision processes in a subject and their average reaction time (RT). Current theory predicts that as computational tasks become more complex, the time required to complete them increases. An analysis of the logic behind model free and model based decision making reveals that model based decisions are much more computationally complex. As such, if the laws governing computational complexity and processing time can be applied to neural circuitry in the same way they are applied to silicon circuitry, subjects who are more model based in their decision making should have slower reaction times.

The current study will address this assertion, as well as the relevant history, theory, and implications, in detail. Section One will begin by giving a brief overview of the field's history, then progress to a

summary of the theory underlying reinforcement learning, as well as general methods by which it can be implemented. Section Two will include a summary of Daw's methodology and results. Section Three contains a description of how we translated the underlying theory into code that constructed parameterized models of each subject's decision making processes, thereby replicating Daw's results. The remainder of the study (Sections Four through Seven) will address our expansion of Daw's results, featuring an analysis of the correlation between the prevalence of model based decision making in a subject and their reaction times.

1.2 History and Origins

The above methods of model free and model based decision making each fall under the broader category of reinforcement learning. The algorithms, techniques, and methods associated with reinforcement learning in modern computational neuroscience stemmed from what were, at the time, considered two very different fields. The first was the study of animal intelligence. Many of the original reinforcement learning techniques and experimental paradigms were developed based on simple trial and error experiments drawn from the field of animal learning. In one of the first scientific texts on the subject, Edward Thorndike discussed his observations of instinctual and habitual animal behavior, paving the way for what would become known in the field as model free learning (Thorndike, 1898).

Half a century later, the same concepts were addressed from a completely different perspective, with a much more abstract approach. Many of the most basic and fundamental equations we use in reinforcement learning stem from this time period. Richard Bellman, father of the Bellman Equation used in Section 4.2 and a prominent computer scientist at the time, laid the groundwork for the field in the 1950's. Bellman's research was inspired by Claude

Shannon's earlier suggestion that "a computer could be programmed to use an evaluation function to play chess" (Sutton and Barto, 1998). Bellman used numeric representations of the current state and available actions, coupled with a value function (or "optimal return function"), to help the computational agent evaluate which action was most likely to result in maximal reward. The new field, christened "dynamic programming" quickly converged with earlier work in animal psychology, giving researchers a more formal framework with which to make and assess predictions about animal behavior and the underlying neural circuitry.

The first rigorous computational investigations of the phenomenon of trial and error learning were conducted simultaneously in 1954, by both Minsky and by Farley and Clark. The term "reinforcement learning" was used for the first time just eleven years later, in Waltz and Fu's 1965 "A heuristic approach to reinforcement learning control systems". Importantly, certain characteristics from both animal psychology and dynamic programming remained. Reinforcement learning, including both model-free and model based cases, must be both selectional and associative. Selectional learning, as Sutton and Barto put it, "involves trying alternatives and selecting among them by comparing their consequences," while associative refers to a process where "the alternatives found by selection are associated with particular situations." (Sutton and Barto, 1998). Selection is closely related to the process of searching, while association is more closely related to remembering. As an example of the boundaries of each, consider the phenomena of natural selection and supervised learning. Natural selection exemplifies selective processes, but is not associative, while supervised learning embodies associative learning, but is not selective. (Sutton and Barto, 1998)

Research continued rapidly through the end of the previous century and the beginning of the current one. Sutton, in a series of papers published in 1978, brought the fields of dynamic

programming and animal psychology closer together than ever by developing a set of learning rules to link changes in predictions of the same quantity made at successive points in time (Sutton 1978), connecting current computational models to past studies done in animal behavior. Major progress in the field was made again by Sutton in 1988, after the use of temporal difference (TD) learning as a general prediction method (Sutton, 1988). The power of reinforcement learning, and more specifically TD learning was highlighted by the success of Tesauro's 1992 "TD-Gammon," a "a neural network that trains itself to be an evaluation function for the game of backgammon by playing against itself and learning from the outcome", (Tesauro, 1992) bringing a new wave of attention to the field and ushering in the modern era of reinforcement learning.

1.3 General Implementations of Reinforcement Learning

There are many different ways to implement reinforcement learning, but all share some basic features and vocabulary. First and foremost, all implementations of reinforcement learning involve the decisions and resultant actions of a computational entity, known as the "agent". Importantly, the tenets of reinforcement learning described above still apply: the agent is not told which actions to take, but instead must make choices that display a balance between a greedy exploitation of the information it currently possesses and an adventurous exploration of the information it is unaware of. The agent must traverse the environment without a teaching signal other than the rewards it receives, which may be spatially or temporally distant from the actions that led to it. Reinforcement learning is not an instance of supervised learning, so the agent must determine how best to optimize rewards by itself

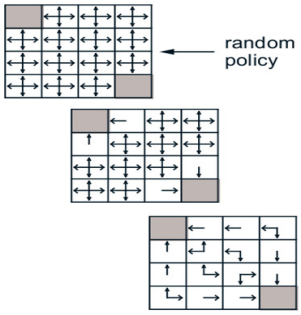

Figure 1.1: A Policy

In this example, the agent's task was to navigate from one of the un-shaded starting cells to either of the two shaded corner cells, without selecting a move that would result in moving off the end of the grid. The state is determined by the current cell the agent is in. Each cell is representative of a different state. Each cell shares the same available set of actions: moving up, down, left, or right. The policy, which in this example is stochastic, is represented by the arrows within the cells. Note how the policy changes over time, moving away from actions which would move the agent off the edge and tending towards actions which would lead towards the rewarded shaded cells. Adapted from Sutton and Barto, 1998

The agent is able to do so by using a mapping of discrete situations, known as "states" to actions to take in those states. This pairing of states and optimal actions is known as a policy (see Figure 1.1). The agent updates its policy after every new action is taken based on both its reward function and its value function. The reward function maps each state to the expected reward to be received upon entering that state. It is essential to reinforcement learning, but is inherently myopic; the reward function can only consider immediate rewards, and cannot look ahead to see what rewards might result from later actions beginning from the current state. The value function corrects the short-sightedness of the reward function by specifying what's good in the long run (see Figure 1.2) (Sutton and Barto, 1998). The value of a state is representational of the total amount of reward an agent can expect to accumulate over the future, beginning from the current

state. Finally, the model's policy, current state, and chosen action can be linked together in what's known as a state-action diagram (see Figure 1.3 for further explanation) (Solway and Botvinick, 2012).

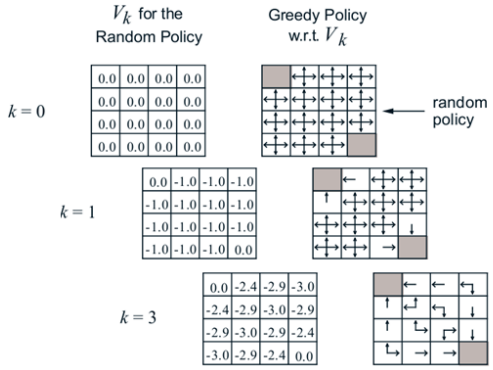

Figure 1.2: A Value Function

The agent must again navigate the same scenario described in Figure 1. The corresponding value function, or the agent's representation of the expected reward from each state, is shown in the left hand grid. The value function is computed by the agent, based on both the transition and reward functions. In this example, the agent receives a reward of -1 for any action in any state. In any state, choosing an action causes the agent to deterministically move one cell in that direction, except those actions which would cause the agent to move off the grid. In those cases, the state of the agent remains unchanged, although the agent still receives a reward of -1 for the action. The values of -3.0 in the center cells reflect that the agent expects it will take three moves to reach the goal state from the current state. The higher than expected values of -2.4 in the cells adjacent to the goal state reflect the random nature of the policy. Even though the goal state may be reached with one action, it may take more than one choice of action before the correct action is selected stochastically. Finally, the value of 0.0 for the goal state reflects that the agent expects it will take zero moves to reach the goal state (i.e. it has already reached the goal state). Adapted from Sutton and Barto, 1998

1.4 Model Free and Model Based Learning

Importantly, some agents include what's known as a "model". The model is an internal representation that mimics the behavior of the environment. When given both a state and an

action, the model would attempt to predict the resultant state and the next reward, after consulting its reward and value functions. Models are therefore the basis of any planning done by the agent. Agents that contain models are termed "model-based", while agents without models are referred to as "model-free".

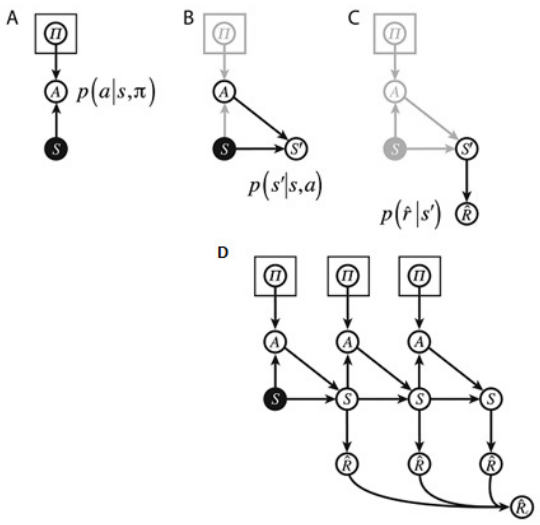

Figure 1.3: A State-Action Diagram

Figure A: Given a state *s* and a policy π with which to evaluate it, the agent can select an action *a*.

Figure B: Given a state *s* and action *a*, the agent finds itself in a new state *s'*.

Figure C: After finding itself in a new state *s'*, the agent receives a reward $\hat{r}$.

Figure D: This process can be iterated over, in each case feeding the new state *s'* generated by the selection of action *a* under policy π to come to a new state *s''*, resulting in a new reward $\hat{r}$. These rewards are summed over all the iterations to produce reward *R*. For all processes described in Figures A-D, the underlying selection mechanics can be either stochastic or deterministic Adapted from Solway and Botvinick, 2012

We will give a quick example to illustrate the differences between model free and model based learning. First, let us consider the model based case. As described above, model based agents have an internal model of the environment, which they can tweak or manipulate to determine the best possible action. Let us consider a driver (we'll call him Bill) who, upon entering a four way intersection, encounters a large pothole across his normal path straight through the intersection. Bill brings up a mental picture of his normal route, and realizes that if he makes a right, two quick lefts, and another right, he will be back on his normal route, beyond the obstruction. Bill's use of an internal representation of the environment (his mental map) makes this a prime example of a model-based decision.

A few moments later, Fred arrives at the same intersection, along his (identical) route to work. Fred, however, makes his decisions in a completely model free way. In past instances where his normal path at this intersection was blocked by huge potholes (this is an especially unlucky intersection), things worked out well when Fred made a right turn. Fred has no idea what will happen when he goes to the right, or what actions he will take once he does, but he is not worried about planning ahead. Fred's lack of plan or internal map of his route, coupled with complete reliance on instinct, help him exemplify the model free decision making process.

2. Nathaniel Daw's Study

Nathaniel Daw recently conducted a study (Daw et al, 2011) to investigate the relative prevalence of the model free and model based agents described above in human decision making. He collected data from 17 participants through 201 trials of a two stage decision task. On each trial, subjects were presented with a choice of two options, each of which was labeled

with a semantically irrelevant Tibetan character. Choice of either of these two options (we'll refer to them as left and right, although throughout the study the same characters appeared on both the left and right sides), led probabilistically to one of two second stage states, which we'll refer to as states 2 and 3. The transition probabilities were arranged so that choosing left in the first (initial) stage lead to state 2 70% of the time (the "common" transition) and state 3 30% of the time (the "rare" transition), while the reverse was true for choosing right (see Figure 2.1). These transition probabilities were fixed throughout the course of the experiment.

Once the initial choice had been made, subjects were presented with an additional choice between two new options, each labeled again with Tibetan characters. The two possible second stage states (state 2 and state 3) were distinguished both by the background color of the choice and by the characters labeling the choices. The subjects were rewarded probabilistically after choosing an option in the second stage. To encourage continued learning throughout the trials, the chance of receiving a reward after choosing a given second stage action was updated after each trial by a random Gaussian walk.

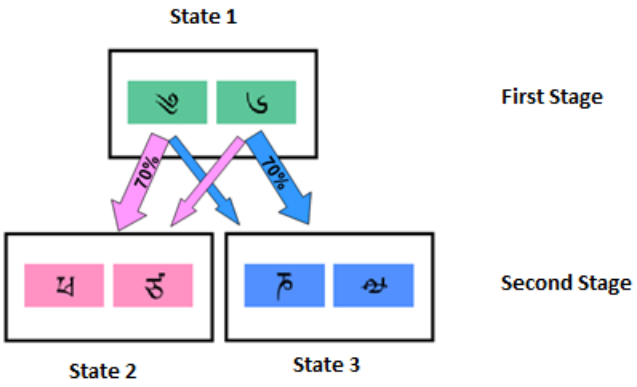

Figure 2.1: Daw's Two Stage Decision Task

> Each trial consisted of two stages. In the first stage, subjects were presented with two choices, labeled by semantically irrelevant Tibetan characters. Choice of the left character (in this depiction) led to State 2 on seventy percent on trials (the "common" transition) and to State 3 on thirty percent of trials (the "rare" transition). The opposite held true for an initial choice of the right character. In the second stage, subjects entered into one of two States (State 2 or State 3), depending probabilistically on their first stage choice. The probability of a reward after a choice in the second stage varied, and was updated after each trial by a random Gaussian walk. (Daw et al., 2011)

At this point, we can begin to see the elegance of the task in its ability to tease out the relative contributions of model free and model based decision processes. The key is to look at the subject's behavior on trials which were ultimately rewarded, where the first stage choice followed the "rare" transition path to the second stage state. On the next trial, a purely model-free agent would display an increased tendency to make the same first stage choice, because that first stage choice was ultimately rewarded on the last trial. However, a model-based agent would make the opposite first stage choice. Although this may at first seem counter-intuitive, upon re-examination it makes sense. The model-based agent is aware of the transition probabilities associated with each of the first states (indeed, that is the entirety of the "model"). In order to

have the best chance of entering the same second stage state it was last rewarded in, it should choose what it knows is the "common" transition path to this state, as opposed to the "rare" transition path it took on the previous trial. The expected stay probabilities on successive first stage choices for model free and model based agents, along with the actual findings from the study, can be seen in Figure 2.2. Special attention is due to Figure C, the report of the patient data from the study, which suggests that human decision making is a mixture of both model free and model based processes.

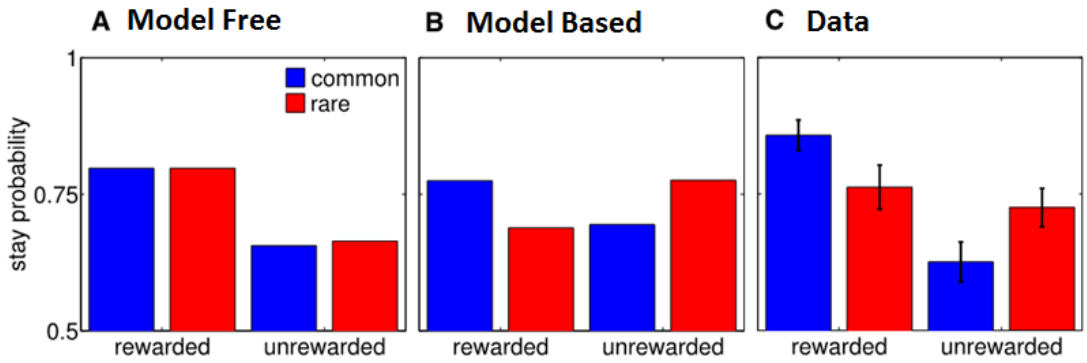

Figure 2.2: Expected and Actual Stay Probabilities

Figure A: Model free agents would be expected to make the same first stage choice after a trial that was rewarded, regardless of whether it entered the second stage via the common or the rare transition path.

Figure B: Model based agents would be expected to take advantage of their knowledge of the environment. If a trial was rewarded after entering the second stage via a common transition path, the agent would be expected to make the same first stage choice again. In contrast, if a trial was rewarded after entering the second stage via a rare transition path, it would be expected to make the opposite first stage choice, in order to maximize the chance of reaching the same second stage state. The same holds true in reverse for unrewarded trials

Figure C: The actual data. Note the results were a mix of both the model free and the model based expectations, suggesting human decision making is a mixture of both processes. (Daw et al., 2011)

3. Construction of Model Free and Model Based Computational Agents

To determine the relationship between the prevalence of model based behavior in a subject (how "model based" they were) and their reaction times, it was first necessary to quantify how model based a given subject's behavior was. This could be accomplished by using the underlying theory to create computational representations of each subject's decision making processes. The process for creating such representations is described in the remainder of this Section. The data used in the below calculations was provided to us by Nathaniel Daw, and was collected during his 2011 study described in the previous Section.

3.1 Overview

It is easy to see that Daw's experimental paradigm lends itself well to the construction of model free and model based agents. First and foremost, the number of possible states and decisions from those states is finite. There are three possible states (one first stage state and two second stage states) and two possible actions from each of those states (left or right) for a total of six possible state action pairs. The value function can therefore be represented as a 2x3 matrix, with one cell for each possible state-action pair. The current example can be compared with the more general case in Figure 1.

As a quick explanation, we will trace the diagram through one hypothetical trial of the study. The diagram begins on the left, with the agent in state $s=1$. After consulting its policy π, the agent chooses an action *a="left."* It finds itself in state $s'=3$, and receives reward $R=0$ (in Daw's experimental paradigm, subjects only receive a reward after a second level choice). The agent then updates its current policy (including the value function described above) to reflect this new information. Next, it will again consult its policy to determine the best action from the current state. This time, the agent chooses action *a="right"* from state $s'=3$ and receives reward $R=1$.

The agent again updates its policy to reflect the new information, and is then returned to state *s=1* for the beginning of the next trial.

The nuances of the above algorithm in model free and model based cases will be discussed in the next Section, but the general process remains the same. The final ingredient for the model is a set of seven basic parameters, each of which describes some aspect of how people update their policies (discussed in more detail below). By combining different values of these parameters with the general process outlined above, it is possible to represent nearly the entire range of decision making strategies.

The next step to find the best fitting set of parameters for each subject. Because decision making is probabilistic in nature, it was necessary find the set of parameters that would maximize the likelihood that the corresponding model would make decisions in a way that was most similar to the subject's actual responses. This was accomplished through the use of two complementary functions. The first (referred to as the constructer) took the seven parameters described above as input, as well as a complete list of the subject's choices and received rewards for each trial. It examined each of the trials in turn, using an iterative algorithm similar to the one described above, and returned an estimate of how likely a fit the model defined by the input parameters was for the subject's actual responses (for full code base, see Appendix B).

The second function (referred to as the driver) was essentially a wrapper around a general purpose numerical optimization function. The driver interfaced directly with the constructor. Through an iterative process similar to a gradient descent, the driver attempted to maximize the likelihood estimate returned by the constructor by tweaking the seven input parameters. (for full code base, see Appendix C). The numerical optimization was conducted a total of ten times for

each subject, from a random set of initial parameters, to ensure it had not settled in a local minimum.

3.2 The Constructor

The constructor consisted of a total of three different computational components -- a model free agent, a model based agent, and a hybrid of the two – which were run in parallel throughout the simulation (see appendix for full code base). Importantly, the hybrid model allowed the relative strength of model free and model based processes to be quantified numerically, through the use of the parameter *w*, which will be described further below. The constructor held the input parameters constant for the full evaluation of the 201 trials, and returned an estimate of the likelihood that the given set of input parameters defined a model that would make decisions in a similar way to the subject's actual responses. The overall parameterization of the models (including *w*) allowed any correlations between the parameters and the subjects' reaction time to be revealed. The input parameters are discussed in turn as they appear below, and are identified by the use of italics.

We will begin by discussing the model free component. Learning at the first level was conducted using the SARSA equation (Sutton and Barto, 1998). In the SARSA equation, the model's estimate of the values of the first stage choices are updated after the second choice is made. The agent calculates the difference between its estimate of the value of the second choice and its estimate for the first choice. It then multiplies this difference by a factor *eta*, known as the learning rate, and adds the product to its estimate of the chosen action in the first state. A similar calculation is performed to determine how to update the value for the second state, except in this calculation the difference is found between the expected value of the second stage choice and the value of the reward received after the choice. It is again multiplied by a factor *eta*, although the

etas for each stage are allowed to vary independently, to better capture differences in learning rates between stages. Finally, the first level values are updated again to reflect the change in estimated values at the second stage. However, this secondary round of first level updates is multiplied by an additional factor *lambda*, known as the eligibility trace, which allows the model to discount more temporally distant influences.

Next, the model based agent is constructed. The agent begins by examining all the previous transitions from the first stage in previous trials to make an educated guess about which first level choice is most frequently associated with each of the two second stage states. It does so using a for loop, examining all trials `1:i-1` (from the first trial to the trial before the current one) and counts the number of time each possible assignment of transition probabilities (i.e. whether choosing left in stage one lead to state two or state three more frequently, indicating which is the "common" transition path). It then updates the value estimates for each of the first stage choices. The agent does so by using the Bellman equation, which in this case consists of a weighted sum of the maximum of the estimated values for the second and third states. For example, if the agent had decided that choosing left in the first state lead to the second state seventy percent of the time, it would update its estimate of the value of choosing left in the first state as follows:

(0.70)(highest valued choice in second state) +
(0.30)(highest valued choice in third state)

The value of choosing right in the first state would be updated in a similar fashion, with the position of the 0.70 and 0.30 weights reversed. Note the absence of *lambda* or *eta* from the first stage of learning in model based agents. The transition probabilities are implicit in the model, so

there's no need to guess at the values of first stage choices, and therefore no need for a factor of *eta*. Similar logic explains the absence of the *lambda* parameter.

Learning in model based agents in the second level is conducted identically to the second level for model free agents. This makes sense, as reward after a second stage choice is assigned probabilistically according to a random Gaussian walk, so there are no transition probabilities to model. Finally, we consider the construction of a hybrid model of the two. Once the other agents have been created, formation of a hybrid is trivial. The values of the hybrid model, for both the first and second stage, is updated on each trial as a weighted sum of the model free and model based agents, governed by a free parameter *w:*

$$Qhybrid = w * Qbased + (1-w) * Qfree;$$

The likelihood of each choice, based on the hybrid model defined by the input parameters, was calculated using the softmax (Daw et al, 2011):

$$P(a_{i,t} = a|s_{i,t}) = \frac{\exp(\beta_i[Q_{net}(s_{i,t}, a) + p \cdot \mathrm{rep}(a)])}{\sum_{a'} \exp(\beta_i[Q_{net}(s_{i,t}, a') + p \cdot \mathrm{rep}(a')])}$$

Beta represents the inverse temperature, and governs how deterministic the model's choices are. Similar to *eta, beta* is allowed to vary between stages to capture any differences between first and second stage choices. Rep(a) is an indicator function that evaluates to one when the first stage choice was the same on the current trial as it was on the previous trial, and evaluates to zero otherwise. The free parameter *p* acts as a weighting mechanism for the indicator function, allowing the model to account for first-order perseveration ($p > 0$) and switching ($p < 0$). After all trials have been examined, the calculated likelihood of the fit was returned to the driver.

4. Expansion of Daw's Research

4.1 Overview

After construction of the models, we conducted an array of statistical analyses on the parameterizations we had found, as well as the subject's reaction times throughout the course of the study. To test our prediction that model based decisions should take longer to complete than model free ones, we examined the ability of *w* to predict reaction times. We also conducted a broader analysis of all computed parameters' ability to predict reaction times and searched for correlations between any of the variables. Importantly, these correlations were unexplored by Daw and his colleagues. One subject's data was removed after an analysis of their model, which displayed an extremely fast reaction time, indicting the subject was not fully engaged in the task. For all subjects, trials where the reaction time was +/- two standard deviations from their mean reaction time were ignored

4.2 Motivation

Computational theory predicts that model based learning would be more computationally demanding than model free learning. This makes sense, and has close analogies to processing and memory access time in computer science. In general, model free decisions require no taxing computations, while model based ones can be incredibly computationally demanding. This is evident both from the code base and from the underlying theory. Think back to the state-action diagram portrayed in Figure 1.3. As it is on the page, the diagram only shows the selected action in each state. Consider what it would look like if the diagram contained all of the possible actions from each state, instead of only the selected one. There would be considerably more action nodes branching away from each state. Now consider if each of those actions lead to a new state, and

new potential actions radiated from each new state. In complex scenarios, the agent must traverse each of these potential routes and select the set of actions that yields the greatest total reward, in a manner similar to a depth-first search. Searches like these are conducted during every use of the Bellman equation, and can be incredibly computationally demanding.

Though Daw's experimental paradigm is constructed to minimize the number of possible ways to traverse the decision task, we can see an analogue to the demanding computations described above present in our code (see appendix). The model based agent requires an additional step compared to the model free agent. The model based agent must update its value estimates using the Bellman equation. In this case, as there are only a maximum two states to look back upon, finding the optimal action is trivial. However, the computational complexity would grow exponentially with each new level added. Even in this trivial form, it is still more computationally demanding than its analogue in the model free agent, which doesn't require a search for the optimal path or any additional computation at the moment of decision. The model free case is closely analogous to accessing a pre-cached value from a computer's memory, where the processor only needs to read a value from memory, but does not need to do any lengthy calculations with it. It is almost always the case that simply reading values from memory and performing basic arithmetic will be faster than accessing an array of values and performing a depth-first search over them[1]. If the neural relationship between computational complexity and

[1] *A note for the computationally inclined:* in computer science terminology, the model free case is said to have complexity O(1), meaning the number of calculations required does not vary with the size of the input (in this case, the number of potential state-action pairs). Model based learning, on the other hand, is much more complex. For purposes of analysis, we will assume state action pairs are represented as nodes of a directed graph, with weights equivalent to the expected value of the state-action pair, and weightless start and end nodes bookending the graph. In the best case, using Djikstra's algorithm with Fibonacci heaps (Fredman and Tarjan, 1984) the model based complexity is O(E + V log(V)), meaning that a computer would be able to find the optimal path among a list of V state-action pairs with E total edges connecting them in E + V log(V) calculations. Importantly, this example is only given as a point of reference. Regardless of how data is stored and paths are calculated, it is unlikely to take fewer computations than E + V log(V), which is significantly more complex than the constant computational complexity observed in the model free case

processing time is analogous to the silicone one, subjects who were model based in their decision making would be expected to have slower reaction times than subjects who were model free.

The above reasoning can be understood in a more concrete way as well. Consider the example of Fred at the intersection mentioned earlier. His model-free decision was quick, and based solely on habit and instinct. Now compare his behavior to Bill's, which was grounded in model-based behavior. When Bill was confronted with an obstacle at the intersection, he conjured a mental picture of his route, and then used basic route planning methods to find a way around the obstacle. Bill must consider all of the possible options, and pick a new route that will work, and hopefully be the fastest. With each new intersection encountered, the number of possible pathways that Bill must consider grows exponentially. Bill's careful; model based planning takes significantly more time than Fred's instinctual decision.

4.3 Supporting Literature

There's some data to support this view in the current literature as well. Otto and his colleagues recently conducted a study investigating the effect of computational load on the relative contributions of model free and model based decision making agents. To do so, they used a task that was functionally equivalent to Daw's two stage decision tree described above (using fractals in place of Tibetan characters), with the inclusion of an additional hundred trials. On one hundred pseudo-randomly selected of these trials, participants were required to complete a Numerical Stroop task while simultaneously completing Daw's decision task. Importantly, the subjects were instructed to focus primarily on the working memory task, and to make choices with "what was left over".

To analyze their results, Otto sorted each subject's trials into three groups, based upon when the most recent Stroop trial had occurred. The trials were denoted Lag-0, Lag-1, and Lag-2, and

referred to cases where the Stroop trial occurred on the current trial, the previous trial, or the trial before the previous trial, respectively. They found a distinct difference in decision making strategies used on Lag-0 trials when compared to Lag-2 trials. On trials with current computational load (Lag-0), subjects did not use their knowledge of the transition structure of the task (indicative of model based thinking), while on normal trials (Lag-1 and Lag-2), subjects used a mixture of model free and model based decision processes, similar to both previous data (Daw, 2011) and our results described in Section Five. Otto's findings imply that computational load affects the way subjects make decisions.

Otto's findings are encouraging for our hypothesis as well. First, they further validate Daw's decision task and experimental paradigm, which was identical to the set-up our study used. More importantly, the results support the computational analyses conducted above. When computational resources were scarce (during Lag-0 trials), subjects tended towards model free decision making, implying model free decisions were less computationally demanding than model based ones. This link is critical; it shows there's a relationship between the theoretical analysis of the complexity of model free and model based learning and how the brain actually responds. In the situations we have examined so far, the brain has responded to problems in ways that are consistent with computational theory. As we have already discussed, theory would predict that model based decisions, because they are more computationally complex, should take longer than model free ones. The next section analyzes that hypothesis in detail.

5. Results

We ran a series of post hoc statistical analyses to assess correlations between the degree to which a subject's decisions were model based and their average reaction time. To conduct the

analyses, we used the R statistics software package, including the ltm (Latent Trait Models) library, as well as the Matlab programming environment. We began by conducting a percentile analysis of the model parameters and log likelihoods of the fits across subjects, which can be seen in Table 4.1. The observed distributions of parameters were consistent with previous analyses, including Daw's own (Daw et al, 2011), suggesting the computational models we created were indeed correct.

	Eta 1st	Eta 2nd	Beta 1st	Beta 2nd	Lambda	W	P	Log Likelihood
25th percentile	0.4483	0.2118	3.298	2.6934	0.4154	0.094	0.0524	167.8667
50th percentile	0.5813	0.42	5.1938	3.6936	0.5595	0.3704	0.1413	197.2618
75th percentile	0.9059	0.7083	7.4349	5.1018	0.9198	0.5221	0.2127	227.5732

Table 4.1: Percentile Analysis of Parameter Distribution

Next, we continued our analysis where Daw's left off by assessing the correlation between *w* and average reaction times at the first level. If model based learning was slower than model free learning, there should be a correlation between the subjects' relative levels of model free and model based decision making and how quickly they responded. No significant correlation or trend was found (see Appendix A, Table 1). Next, to better standardize first level reaction times across subjects, a multiple regression analysis was conducted with the inclusion of all parameters in the model (see Appendix A, Table 2). No significant correlation was found between *w* and first level reaction times even after the inclusion of all other parameters,

although a positive trend between *eta* and the second stage and first level reaction times was revealed. The above analysis was repeated, additionally allowing for interactions between *w* and all parameters uniquely used in model free learning, with similar results (see Appendix A, Table 3).

To better disentangle the underlying relationships between the parameters and further explore the link between *eta* at the second level and reaction time, additional multiple regression analyses were conducted between various subsets of the model parameters and first stage reaction time (not included). In almost all cases, the observed relationship between second level *eta* and first level reaction times was confirmed. All regressions conducted displayed a positive relationship between first level RTs and *w*, though few were significant. One regression was especially of interest, due to the observed trend between *w* and first stage RT's, and is included below.

We examined the combined ability of *w, eta* at the first level, *eta* at the second level, and *beta* at the second level to predict the subject's average reaction time at the first level:

$$(rt1 \sim w + eta1 + eta2 + beta2)$$

The overall correlation between the first level reaction times and the above parameters was found to be significant, with $p < 0.05$. The *eta* parameters at both the first and second stages were found to be the most significant contributors to the regression, with $p < 0.01$ and $p < 0.05$, respectively. Importantly, there was a trend observed between *w* and first level reaction times, with $p < 0.10$. For a full summary of the regression analysis, see Table 4.2

	Estimate	Std. Error	t value	Pr(>|t|)	Significance
(Intercept)	583.328	59.109	9.869	8.44E-07	***
w	123.931	60.2	2.059	0.06402	x
Eta 1st	-135.071	57.283	-2.358	0.03795	*
Eta 2nd	185.656	57.789	3.213	0.00827	**
Beta 2nd	16.121	8.103	1.99	0.07208	x

Significance Codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 'x' 0.1 ' ' 1

Table 4.2: Multiple Regression Analysis

Finally, pairwise correlation analyses were conducted for all model parameters, as well as first and second level reaction times. The tests revealed a few interesting relationships. There was a significant negative correlation observed between the first level inverse temperature parameter *beta* and the decay rate of the eligibility trace *lambda*, with $p < 0.01$. We also observed a significant positive correlation between first level *beta* and the free parameter w, which governed the relative contributions of each type of model, with $p < 0.01$. The second level *beta* and *p*, the weighting coefficient of the indicator function that detected whether the same first level choice was made on the previous trial were significantly positively correlated as well, with $p < 0.01$. Finally, the negative correlation between *lambda* and *w* was observed to be significant, with $p < 0.01$. For a full summary of the results, see Table 4.3

	Eta 1st	Eta 2nd	Beta 1st	Beta 2nd	Lambda	W	P	RT 1	RT 2
Eta 1st		0.419	0.084	-0.281	-0.488	0.424	-0.153	-0.167	-0.303
Eta 2nd	0.106		0.05	-0.26	-0.029	0.267	-0.003	0.472	0.082
Beta 1st	0.758	0.854		-0.285	-0.703	0.723	-0.274	0.224	-0.12
Beta 2nd	0.291	0.33	0.284		0.504	-0.413	0.653	0.203	0.357
Lambda	0.055	0.915	**0.002**	0.046		-0.746	0.419	0.164	0.357
W	0.102	0.318	**0.002**	0.111	**0.001**		-0.241	0.242	-0.045
P	0.571	0.991	0.305	**0.006**	0.106	0.368		0.446	0.448
RT 1	0.536	0.065	0.405	0.45	0.544	0.367	0.083		0.403
RT 2	0.254	0.763	0.659	0.175	0.175	0.869	0.082	0.122	

Table 4.3: Correlations between Model Parameters and Reaction Time

Correlation analyses were conducted between all possible pairwise combinations of the parameters used to define the model, as well as first and second stage average reaction times. The top diagonal lists the calculated correlation coefficients, while the bottom triangle includes the corresponding p-values. Correlations that were significant ($p < 0.01$) have been bolded and highlighted in blue.

6. Discussion

6.1 Correlations with Reaction Time

While it was not significant, the trend between *w* and reaction time observed in the selected multiple regression analysis (Table 4.2) is encouraging. It provides some evidence that, even while only considering this relatively small sampling of subjects, there is some relationship between how model based a participant's decisions are and how quickly they responded. However, for the purposes of full disclosure, it is necessary to reiterate that this regression represented only one of many similar analyses that were conducted. With each new test, the

probability of a Type I error increases. The trend is encouraging, but further testing and more rigorous analysis are necessary for it to be significant.

There are a number of possible explanations why no strong correlation between *w* or other parameters of the model and the first stage reaction time was found. First and foremost, we only considered each subject's data and associated models based upon the study as a whole. The models were constructed based on all 201 consecutive trials. However, one might expect differences in the models at different points throughout the 201 trials. For example, there might be learning period at the beginning of the study for each subject, where their responses were more model free. Similarly, if at a particular point in the study a subject identified a particular second level choice to be particularly rewarding, their responses might be more model based as they attempted to seek out that trial. We were unable to consider what the model looked like at the time the decision was made. We also were unable to consider how the parameters changed throughout the course of the study. Each of these may have had some relationship with RT. By constructing the models based on the subjects' data as a whole, and averaging their reaction times, it is possible that more nuanced relationships contained in the data were missed.

The weak correlation between the model's parameters and reaction time may also point towards the pre-caching of updated values for state-action pairs. It is possible that subjects computed an updated estimate of the value for the most recently visited state and corresponding action immediately after the action was made, instead of the next time the value was needed. By precompiling the values in this fashion and storing them for later, the models could make model based decisions as quickly as model free ones.

However, the Otto study, by demonstrating an increased preference for model free learning on trials where processing was interfered with *at the time of decision making*, implies that values are updated as needed, instead of immediately after decision making. This seems to contradict the theory of precaching, where values are updated immediately after decisions are made and then stored for later. However, the Stroop task used in the Otto study was much more computationally demanding than Daw's decision tree. Furthermore, subjects were instructed to focus on the Stroop task, leaving Daw's decision tree to be of secondary importance. It is possible, especially when subjects gave their full attention to the Daw's computationally simple decision tree, that subjects were able to update the expected values of the previous state while simultaneously making a decision in the next state. In other words, Daw's task may not have been computationally demanding enough to require subjects to switch to model free decision making. If this were the case, the updated values would be cached for future use at the same time as the agent was making its decision, minimizing differences in reaction time between model free and model based processes.

The reaction times may also be moderated by other effects that we did not account for. One likely candidate is the difficulty of discrimination between choices at the next stage. In either case, the model updates its estimated values of the previous stage based on the expected value for the next stage. The values of the first stage are updated according to the model's estimate of the choices it has in the second stage, and the values of the second stage are similarly updated based on the rewards it expects after each choice. If one choice at either level is obviously the better of the options available, it should be easy for the agent to make a decision, resulting in a quick reaction time. However, if the values for each of the choices are similar to each other, it

may be more difficult for the agent to choose between them. In this case, one would expect to see a slower reaction time, as the agent reasons through which option is the better of the two.

Finally, there remain two relatively simple explanations for the lack of observed correlation, each of which stems from the design of the study. One of the simplest explanations could be a result of the small sample size. After removing one subject as described in Section 4.1, only sixteen subjects remained. If the reaction time effect size were subtle, or moderated by other aspects of the study, it could be difficult to detect. Another possible explanation could be effects of subject fatigue throughout the study. Participants were required to make a total of 402 decisions throughout the course of the study, excluding the training tasks. The study itself took an average of nearly twenty minutes to complete. It could be difficult to focus attention on a repetitive task for long durations of time. The study included only minimal analysis to identify and remove potential trials during which the subject was inattentive.

6.2 *Eta* at the Second Level

Eta at the second level appeared in many of the multiple regression analyses we conducted as a significant predictor of first level reaction times. The observed correlation was always positive, indicating that subjects with higher learning rates at the second stage tended to respond more quickly on the first stage. This relationship is present as a trend in the pairwise analysis between *eta* and first level reaction times as well.

This is likely to be the result of the close proximity of second level decisions and first level responses on the subsequent trials. It could be indicative of some necessary post-processing that occurs immediately after the second level choice has been selected, which would occur during the begging of the first level reaction period of the next trial. It is also reflective of the

structure of the code. Second level *etas* were one of the few parameters that affected all aspects of the code. Both model free (through the secondary updates of the first level values in the first stage) and model based (through the use of the Bellman equation in the first stage) decision making processes were influenced by second level *etas*, as well as both the first and the second level updates. After realizing how far reaching second level *eta* values were, it's not surprising that it appeared as a significant predictor in many of our analyses.

6.3 Pairwise Analyses

There were a number of significant correlations found between the parameters in the model. The parameters are unrelated to the reaction times, but are instead interesting based on what they reveal about the decision making process. Each is reflective of some aspect of how subjects made decisions, and also of the underlying structure of the code driving creation of the models. We will address each in turn, beginning with those involving the first level inverse temperature parameter *beta*.

Beta governs how probabilistic people's decisions were. A subject with *beta* equal to zero would be uniformly random in their decision making, while a subject with a higher *beta* would be much more deterministic. This explanation can help us interpret the negative correlation between *beta* and *lambda*. *Lambda* directly controls the magnitude of the secondary updates of the first level values, to reflect the latest changes in the second level after second level decisions have been made. For subjects who were more random in their first level decisions (smaller *beta*), these new updates were critical. To continue to maximize reward while making random decisions, subjects must continue to learn. These strong updates (large *lambda*) may be indicative of the relationship between random decision making and the rapid changes in value estimates required after making such decisions.

The correlation is reflective of the structure of the code as well. The *lambda* parameter is only used to update values in model free agents at the first stage, so it makes sense that two parameters associated only with the first stage were correlated. The same approach can be used to understand the negative correlation between *w* and *lambda.* As described above, *lambda* is only used in model free agents, so it makes sense that as people become more model based, there's less dependence on lambda as an update mechanism.

Similar logic can be used to understand the positive correlation between *w* and *beta* at the first level. As subjects became more model free in their decision making (decreased *w*), they became less deterministic. This fits with the theory of reinforcement learning. Exploration is critical for model free learning. The reverse is true as well. Subjects who were model based in their decision making would have a better understanding of their environment, and therefore it may be preferable for them to focus on exploitation instead of exploration at the first stage. There may be an effect due to the structure of the code, similar to the one described above, as well. Model free and model based agents differ only in the first stage, so it is not surprising that there was a correlation between the first *beta* and *w*.

We can consider the correlations between *lambda* and *beta,* as well as between *beta* and *w* on a higher level as well. The positive correlation between *w* and *beta* at the first level indicates that subjects who were more model free in their decision making were more random in their choices. The negative correlation between *lambda* and *beta* suggests that subjects who were more random in their choices were more likely to use a "Monte Carlo" approach (high lambda) to decision making, where information about the environment is gathered by repeated, random sampling. Taken together, these two relationships may give an indication of subjects' overall level of participation in the study. Those who were actively engaged in the task may have made more

model based, deterministic choices, while those who were not engaged may have made more random, model free decisions using the Monte Carlo approach.

Finally, we will consider the correlation between the second level *beta* values and the *p* coefficients. The *p* coefficient was representative of how much additional weight was given to first stage choices where the same action was selected in the first stage on the previous trial. Subjects who were more deterministic in their second stage choices tended to give more weight to repeated choices. This may be reflective of a belief in certain subjects that they had "figured out" the task, or understood which first stage choice lead predominantly to which second stage state. Subjects like these gave special attention to tasks where the choice was the same in the first stage as it was in the last trial, and were focused in their second stage decisions. These subjects appeared to be targeted in their decision making.

7. Suggestions for Further Study

There are a number of potential avenues for further research using the experimental paradigm Daw has developed (or slightly modified versions.) To address potential issues raised in the discussion, the study should be expanded to include more participants. An algorithm, operating both during and after the study has been completed, should be put in place to detect subjects who may be suffering from the fatigue effect. This algorithm could rely on on-line updates of the *beta* parameters; if subjects became too random in their decision making (low *beta,* or rapid fall of *beta*), the test could request they re-focus on the study. Alternatively, modern cycle detection analysis (see Shmueli, 1982) could be used to detect repetitive patterns in subject responses and then make similar requests.

Additionally, further analysis of the current data should be conducted to determine correlations between reaction time and model parameters on more finite scale. Multiple models

could be constructed for each subject, using blocks of twenty five or more trials per model. This would provide a finer-grain picture of the parameters at any given point in the trial, as well as how they changed over time, allowing for a more nuanced investigation of the underlying relationships between the parameters and reaction times. The use of averaged reaction times, even after extreme trials are removed, may not detect some of the more subtle relationships hidden in the data.

It would also be of interest to vary the allowed response time (making sure to instruct subjects of the change) or to set a minimum response time for subjects. Decreasing the time subjects had to respond may promote more model free behavior, while setting a minimum reaction time may exclude instinctual decisions. A follow up study could place subjects into one of three groups: a group where subjects were forced to respond under ~750 milliseconds, a group where subjects were required to give a response in between ~750 and ~1750 milliseconds, and a group where subjects were forced to wait a minimum of ~1750 milliseconds before responding. The study could then examine whether any differences occurred in the relative contributions of model free and model based agents between time-restricted groups

More neurobiological approaches should be considered as well. Recent literature (Hampton, Bossaerts, and Doherty, 2006) points to the ventromedial prefrontal cortex (vmPFC) as a potential substrate for model based decision making. The use of transcranial magnetic stimulation (TMS) to selectively inactivate the vmPFC, coupled with Daw's decision making paradigm, would provide a way to quantitatively evaluate this hypothesis. TMS could be coupled with the grouped RT described above, to better understand the link between the vmPFC, reaction times, and the prevalence of model based learning.

Bibliography

Alexander, G., and M. Crutcher. "Functional Architecture of Basal Ganglia Circuits: Neural Substrates of Parallel Processing." *Trends in Neurosciences* 13.7 (1990): 266-71.

Cave, Kyle R., and Jeremy M. Wolfe. "Modeling the Role of Parallel Processing in Visual Search." *Cognitive Psychology* 22.2 (1990): 225-71.

Daw, Nathaniel D., Samuel J. Gershman, Ben Seymour, Peter Dayan, and Raymond J. Dolan. "Model-Based Influences on Humans' Choices and Striatal Prediction Errors." *Neuron* 69.6 (2011): 1204-215.

Farley, B., and W. Clark. "Simulation of Self-organizing Systems by Digital Computer." IEEE Transactions on Information Theory 4.4 (1954): 76-84.

Fredman, Michael, and Robert Tarjan. "Fibonacci Heaps and Their Uses in Improved Network Optimization Algorithms." Journal of the Association for Computing Machinery 34.3 (1987): 596-615.

Klopf, Harry A. "Brain Function and Adaptive Systems - A Heterostatic Theory." *Air Force Cambridge Research Laboratories* (1972).

Loewenstein, G., and T. O'Donoghue. "Animal Spirits: Affective and Deliberative Processes in Economic Behavior." *Cornell University, Center for Analytic Economics* (2004).

Minsky, Marvin. "Neural Nets and the Brain Model Problem." Princeton University Ph.D Dissertation in Mathematics (1954).

Otto, A. R., Samuel J. Gershman, Nathaniel D. Daw, and Arthur B. Markman. "Dissecting Multiple Reinforcement Learning Systems by Taxing the Central Executive." *University of Texas at Austin* ([2012]).

Rauschecker, Josef P. "Parallel Processing in the Auditory Cortex of Primates." *Audiology and Neuro-Otology* 3.2-3 (1998): 86-103.

Everitt, Barry J., and Trevor W. Robbins. "Neural Systems of Reinforcement for Drug Addiction: From Actions to Habits to Compulsion." Nature Neuroscience 8.11 (2005): 1481-489.

Rummery, G. A., and M. NiranJan. "On-Line Q-Learning Using Connectionist Systems." Cambridge University Engineering Department (1994).

Shmueli, Oded. "Dynamic Cycle Detection." Information Processing Letters 17.4 (1983): 185-88

Solway, Alec M., and Matthew Botvinick. "Goal-Directed Decision Making as Probabilistic Inference: A Computational Framework and Potential Neural Correlates." Psychological Review 119.1 (2012): 120-54.

Stone, Jonathan, Bogdan Dreher, and Audie Leventhal. "Hierarchical and Parallel Mechanisms in the Organization of Visual Cortex." *Brain Research Reviews* 1.3 (1979): 345-94.

Sutton, Richard S., and Andrew G. Barto. *Reinforcement Learning: An Introduction*. Cambridge, MA: MIT, 1998.

Sutton, Richard S. "Learning to Predict by the Methods of Temporal Differences." *Machine Learning* 3.1 (1988): 9-44.

Sutton, Richard S. "Single Channel Theory: A Neuronal Theory of Learning." *Brain Theory* 4 (1978): 72-75.

Tesauro, Gerald. "Temporal Difference Learning and TD-Gammon." *Communications of the ACM* 38.3 (1995).

Thorndike, E. L. "Animal Intelligence: An Experimental Study of the Associative Processes in Animals." *Psychological Review* 5.5 (1898): 551-53.

Waltz, M., and K. Fu. "A Heuristic Approach to Reinforcement Learning Control Systems." *IEEE Transactions on Automatic Control* 10.4 (1965): 390-98.

Acknowledgements

I would like to thank Professor Matthew Botvinick and Alec Solway for their advice, guidance, and discussions throughout the course of the study

I would like to thank Professor Yael Niv, Carlos Diuk and Angela Radulescu for their feedback and assistance while both writing and debugging the code

I would like to thank Carrie and Len Grabowski, and Rachel Naar for their help and thoughtful comments while proofreading the report, as well as for putting up with me during the process.

Appendix A

Supplementary Tables

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	-0.2343974	0.670585	-0.35	0.732
RT1	0.0008934	0.0009589	0.932	0.367

Table A.1: Correlation between w and RT1

	Estimate	Std. Error	t value	Pr(>|t|)	Signif
(Intercept)	524.2	120.567	4.348	0.00245	**
eta1	-92.354	75.636	-1.221	0.25684	
eta2	146.173	68.493	2.134	0.06537	x
beta1	5.702	7.617	0.749	0.47551	
beta2	3.337	11.366	0.294	0.77656	
lambda	75.921	114.729	0.662	0.52673	
w	107.374	95.883	1.12	0.29527	
p	208.94	143.906	1.452	0.18458	

*Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 'x' 0.1 ' ' 1*

Table A.2: Correlation between Model Parameters and RT1

	Estimate	Std. Error	t value	Pr(>|t|)	Signif.
(Intercept)	503.644	205.893	2.446	0.0582	x
w	226.497	488.656	0.464	0.6625	
eta1	-74.744	156.736	-0.477	0.6536	
eta2	156.167	124.282	1.257	0.2644	
beta1	0.775	11.561	0.067	0.9492	
beta2	4.752	16.261	0.292	0.7819	
lambda	99.753	171.241	0.583	0.5855	
p	286.181	230.031	1.244	0.2686	
w:eta1	32.475	397.671	0.082	0.9381	
w:lambda	-153.33	602.135	-0.255	0.8091	
w:p	-473.451	823.391	-0.575	0.5902	

*Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 'x' 0.1 ' ' 1*

Table A.3: Correlation between Model Parameters and RT1, with the inclusion of possible interactions with w

Appendix B

Hybrid Model Construction

```
function LL = rllik_hybrid(eta1, eta2, beta1, beta2, lambda, w ,
     p, state,choice1,choice2,money)

%
% Adapted from code provided at seminar by Professor Yael Niv
%
% LL = rllik(eta,beta,lambda,state,choice1,choice2,money)
%
% output:
% LL - the log likelihood of the data
% input:
% eta1/2    - learning rate, first level/second level
% beta1/2   - softmax inverse temperature, first level/second
     level
% lambda    - eligibility trace decay rat (set to 0 to get TD(0)
     without eligibility traces)
% state     - 1 is the top level, 2 and 3 are the bottom level
% choice1   - the choice at the top level -- 1 or 2 (0 for missed
     trials)
% choice2   - the choice at the bottom level -- 1 or 2 (0 for
     missed trials)
% money     - amount won (1 or 0)
% w         - percent model free/model based
% p         - weighting  for consecutive identical first stage
     choices

NStates = 3;
NActions = 2;
NTrials = length(choice1);

Qfree = zeros(NStates,NActions); % initialize Qfree values to 0
Qbased = zeros(NStates,NActions); % initialize Qbased values to 0
Qhybrid = zeros(NStates,NActions);
LL = 0 ;                         % initialize log likelihood
prev = 0;
% main loop
for t = 1:NTrials
    repa = zeros(1,NActions);

    % stop if trial was missed
    if (choice1(t) == 0 || choice2(t) == 0)
        continue
    end

    other = 0;
    if choice1(t) == 1
        other = 2;
    end
```

```
if choice1(t) == 2
    other =1;
end
if other == 0
    ff('bad first level choice, choice = %d\n', choice1(t))
    continue
end


%assign p
if t>1
    if choice1(t-1) ~= 0
        repa(choice1(t-1)) = 1;
    end
end


% first level choice likelihood
LL = LL + beta1*(Qhybrid(1,choice1(t)) + p *
 repa(choice1(t))) - logsumexp(beta1*(Qhybrid(1,:) + p *
 repa));

% second level choice likelihood
LL = LL + beta2*Qhybrid(state(t),choice2(t)) -
 logsumexp(beta2*Qhybrid(state(t),:));

%first level model free
PE = Qfree(state(t),choice2(t)) - Qfree(1,choice1(t)); %SARSA
Qfree(1,choice1(t)) = Qfree(1,choice1(t)) + eta1*PE;

% second level model free
PE = money(t) - Qfree(state(t),choice2(t));

%Update second level model free
Qfree(state(t),choice2(t)) = Qfree(state(t),choice2(t)) +
 eta2*PE;

%Update 1st level again model free
Qfree(1,choice1(t)) = Qfree(1,choice1(t)) + eta1*PE*lambda;

% learning at second level model based
PE = money(t) - Qbased(state(t),choice2(t)) ;

%Update second level model based
Qbased(state(t),choice2(t))  = Qbased(state(t),choice2(t))  +
 eta2 * PE;

%initialize count parameters
left = 0;
right = 0;


    %determine which transition occurs more frequently, and thus
     receives
```

```
    %the higher weighting
    for i = 1:t
        if choice1(i) == 1 && state(i) == 2
            left = left + 1;
        end
        if choice1(i) == 2 && state(i) == 3
            left = left + 1;
        end
        if choice1(i) == 1 && state(i) == 3
            right = right + 1;
        end
        if choice1(i) == 2 && state(i) == 2
            right = right + 1;
        end
    end

    %assign weighting based on most frequent choice
    if left > right
        weight2 = 0.7;
        weight3 =0.3;
    else
        weight2 = 0.3;
        weight3 = 0.7;
    end

    %first level model based
    %Bellman Equation
    Qbased(1,1) = weight2 * max(Qbased(2,:)) + weight3 *
     max(Qbased(3,:));
    Qbased(1,2) = weight3 * max(Qbased(2,:)) + weight2 *
     max(Qbased(3,:));

    % hybrid update
    Qhybrid = w * Qbased + (1-w) * Qfree;

end

% we are minimizing this function, so use minus LL

LL = -LL;
```

Appendix C

Hybrid Model Driver

```
%
% Adapted from code provided at seminar by Professor Yael Niv
%

%clear all; clc;

Subjects = [17];
Nsubjects = 17;

C1 = []; C2 = []; R = []; S = []; subj = []; react1 = []; react2
     = [];

% loading the subjects' behavioral data
SubjFile = dir('dawdatatrans.mat');

% parsing the subjects' behavioral data
for s = 1:Nsubjects
     offset = ((s-1) * 201) + 1;
     endoffset = offset + 200;
     C1 = [C1; ch1(offset:endoffset)]; % the choices at level 1
     C2 = [C2; ch2(offset:endoffset)]; % the choices at level 2
     R  = [R; mn(offset:endoffset)];     % the rewards
     S  = [S; st(offset:endoffset)];     % the states at level 2
     react1 = [react1; rt1(offset:endoffset)]; %first stage RT
     react2 = [react2; rt2(offset:endoffset)]; %second stage RT
end

[Nsubjects,Ntrials] = size(S);

optset = optimset('algorithm', 'sqp', 'Display', 'off');
Fit = {};
clear Eta1 Eta2 Beta1 Beta2 Lambda w p
for iter = 1:10;    % run 10 times from random initial conditions,
     to get best fit
    for i = 1:Nsubjects;
        ff('%d...',i)

        LB = [1e-6 1e-6 1e-6 1e-6 1e-6 1e-6 -10];
        UB = [1-(1e-6) 1-(1e-6) 20 20 1-(1e-6) 1-(1e-6) 10];
        init = rand(1,length(LB)).*(UB-LB)+LB; % random
     initialization within the bounds

        % finding the minimum of the function rllik
        [res lik] = fmincon(@(x)
     rllik_hybrid(x(1),x(2),x(3),x(4),x(5), x(6), x(7),
     S(i,:),C1(i,:),C2(i,:),R(i,:)),init,[],[],[],[],LB,UB,[],...
            optset);

        % gathering results
```

```
        Eta1(i) = res(1);
        Eta2(i) = res(2);
        Beta1(i) = res(3);
        Beta2(i) = res(4);
        Lambda(i) = res(5);
        w(i) = res(6);
        p(i) = res(7);
        Lik(i) = lik;
    end
    ff('\n')

    Fit{iter} = [[1:Nsubjects]' Eta1' Eta2' Beta1' Beta2' Lambda'
     w' p' Lik'];
    L(:,iter) = Lik';  % Check this to see the likelihoods from
     the different runs (to check how stable the fits were to
     different starting points)
end

% find the best fit of all 10 runs
clear BestFit
[a,b] = min(L');
for i = 1:Nsubjects
    BestFit(i,:) = Fit{b(i)}(i,:);
end

%  the results
ff('Sub\t eta1\t eta2\t beta1\t beta2\t lambda\t w\t p\t LL\n')
ff('%d\t %3.3f\t %3.3f\t %3.3f\t %3.3f\t %3.3f\t %3.3f\t%3.3f\t
     %3.3f\t\n',BestFit')
```