Resource Rational Analysis of Memory Use in Bayesian Updating

Lucy Li (lucy3@stanford.edu)
Symbolic Systems Program
Stanford, CA 94305 USA

George Pakapol Supaniratisai (sspkpl@stanford.edu)
Department of Computer Science
Stanford, CA 94305 USA

Stephanie Zhang (szhang3@stanford.edu)
Symbolic Systems Program
Stanford, CA 94305 USA

Abstract

Bayesian inference has been used to model many aspects of cognition, and this paper aims to investigate memory using a computational perspective. If memories are costly, how should a learner allocate its resources and still make accurate predictions? We present a framework for an agent that seeks to learn the true weight of a coin after observing a fixed number of observations. The agent pays to retain a noise-minimal belief and increase prediction accuracy. The beliefs of the agent reside in a hierarchical structure, and we hope to see how different types of information should be managed using a resource-rational strategy.

Keywords: memory, forgetting, Bayesian models, computational models, resource rationality, hierarchical models

Introduction

"The strength of your memory dictates the size of your reality." (Klosterman, 2003) Our ability to draw from past experiences to make inferences about the world guides the way that we interact with our environment. In this sense, memory is one of the most powerful tools of human functioning. However, anyone who has experienced the frustration of the 'tip of the tongue effect' knows that memory is not a perfect system. Due to limited resources, stored information experiences degradation, intrusion errors, and corruption. All natural learners are non-ideal ones, and ensuring perfect retrieval is costly. This all leads us to one critical question: what is the optimal way to handle limited memory bits while retaining predictive power?

Background

Resource Rationality

Since Hermann Ebbinghaus pioneered the first formal examination of memory in 1858, researchers have observed the effects of many variables on memory retention. These variables include but are not limited to attention, exposure, time since encoding, and cognitive state similarity between encoding and recall (Chun & Turk-Browne, 2007; Anderson & Schooler, 1991; Murre & Dros, 2015; Spiro, 1980).

A general theme in memory research is that humans have limited cognitive resources, and must make decisions—either consciously or unconsciously—about what information to maintain, and what to discard. When presented with new information in an environment, an individual must consider both the predictive power of that information, but also the cognitive costs of storing it.

To conceptualize this, consider the following thought experiment. Suppose that you observe a series of coin flips across a number of days, and aim to predict the next few flips on the final day. After each observation, you generate a posterior belief about the weight of the coin, and relay this information to your scribe, who will record your belief. While your scribe will inevitably make mistakes, the more you pay him, the harder he will try, and the less mistakes he will make when recording your belief. At the end of the day, your scribe returns back your posterior belief with errors, and this distribution becomes your new prior for the next day's observation.

In the context of this problem, you can think of your scribe as your memory. What you 'pay' your memory may be attention, time, or retrieval practice. However, these activities have a cost, and your resources for them are limited. The objective, therefore, is to determine how to allocate these resources in order to maximize predictive power.

The field of study that seeks to model this decision process is known as *bounded* or *resource rationality* (Simon, 1957). Resource rationality presents a working framework with which to think about optimization—especially when learning is viewed as a sequence of observations. As stated by the statistics maxim, 'Today's posterior is tomorrow's prior,' each observation updates the current posterior belief to become the new prior belief. However, in the case of memory, today's prior belief is only a noisy version of yesterday's posterior belief. In our current model, we attempt to apply resource rationality to investigate this information loss in Bayesian inference.

Hierarchical Models

Psychology research suggests that we do not merely remember information in terms of bits. Rather, findings from Rosch and colleagues suggest that semantic memory is organized in complex networks and hierarchies (Rosch, Mervis, Gray, Johnson, & Boyes-Brahem, 1976). For example, consider how 'dog' may fit in a hierarchical knowledge structure. At the *subordinate level*, an 'dog' object can be classified as a 'corgi' and at the *superordinate level*, it can be categorized as 'animal'. As we go down in this hierarchy, each level reveals a new layer of specificity. Alternatively, as we approach higher levels in the hierarchy, the level of abstraction increases.

Given this hierarchical structure, abstract knowledge contained at the higher levels can guide the learning of more specific information. For example, knowing that a 'corgi' is a subclass of 'dog' allows any object classified as 'corgi' to inherit the probabilities of properties given at the 'dog' level. So, information at higher levels of abstraction can still be useful for making inferences and predictions about one's environment. Thus, it is important to consider the cost of remembering information at different hierarchical levels.

Related Works

There has been some research on how memory affects serial reproduction, using a Bayesian model (Xu & Griffiths, 2010). The reconstruction of a stimulus such as a picture is passed from one participant to another, and the study saw that serial reproduction converges to the prior that results from training, which means memory biases skew participants' reproduction of observed stimuli.

Previous research in resource rationality suggests that individuals make optimization decisions according to a utility function, where predictive power is maximized and cost is minimized. While this describes the behavior of an ideal Bayesian agent, it fails to capture the behavior of a standard learner. Instead, recent research shows that individuals do not make predictions based on the full posterior probability distribution, but rather base decision on samples from this distribution. In order to address the question of what sampling theory best optimizes reward, Vul and colleagues showed that locally suboptimal sample-based approximations may be the globally optimal strategy over long periods of time (Vul, Goodman, Griffiths, & Tenenbaum, 2014).

While previous research asked *how many* samples an individual should accumulate to optimize reward, we hope to define some other conditions that parallel imperfect memory, and show that optimizing overall gain does not require paying as much as possible.

Experiment 1: Methods

Overview

We begin by exploring a single-hierarchy case, where we built a model that infers the weight of a coin when given a sequence of true and false flip outcomes as observations. This, along with our successive models, were created in the probabilistic programming language WebPPL developed by Goodman and Stuhlmüller (2014)[1].

We construct our model recurrently, that is, the posterior belief of the coin weight after observing n data points becomes the prior belief before observing the $(n+1)$th data point. This paradigm is often used for models of learning as conditional inference (Goodman & Tenenbaum, n.d.). For an example, suppose we have a total of two observations. We update the prior after making one observation d_1, and update again after making another observation d_2. The posterior after d_1 is the prior for d_2. This is equivalent to updating the prior after making a batch observation of both d_1 and d_2:

$$p(h|d_1,d_2)p(h) \propto p(h|d_2)p(h|d_1)p(h), \tag{1}$$

where h is the hypothesis or belief about the weight of the coin and d_1 and d_2 are two independently generated data points, or flips of the coin.

To model an imperfect learning processes, we introduce a noise function that is applied between each observation, which in the above simple case would be between d_1 and d_2. Thus, while an ideal model has the previous day's posterior as the next day's prior, a non-ideal model has a noisified posterior as the next day's prior.

In this experiment, the amount of noise applied in the model decreases when the learner makes a higher payment, or effort cost. At the end, the learner predicts the next k flips. The model will receive less penalty if it predicts a closer number of head flips to the actual number of flips. This penalty is calibrated to scale well with the magnitude of payment.

Our goal is to see how a rational agent should pay to reduce noise while achieving the optimal error-payment tradeoff. We expect to see that it may not always be optimal to pay for the full posterior between making one observation and another.

For each of our scenarios, which differ by the noise function used, we ran the model 100 times and averaged the performance results.

Environment parameters

We have several adjustable parameters in our environment, listed as follows:

The number of time-steps of observations, n where $n \in \mathbb{Z}^+$ and $n \geq 2$. These observations are generated by repeatedly flipping a coin with a true weight, which the model aims to learn. Any string of observations is possible by any coin, so it is important to note that a more stereotypical sequence of observations makes the weight easier to learn. For example, 19 trues and 1 false has a higher probability of being generated by a coin with 0.9 weight than 10 trues and 10 falses, so a learner that observes the former will be better at learning the true weight than a learner that observes the later.

Our models have this parameter set to 20.

Coin weight. In our experiment, our coin is biased, e.g. with a probability of head flips $p_h = 0.8$, so that the model has something to learn.

Our prior belief for the weight of the coin is set to be a logit-normal distribution, where $\mu = 0$ and $\sigma = 1.5$, with a lower bound of 0 and an upper bound of 1. This distribution looks much like a uniform one.

Calibration factor. In this experiment, performance is based on how our model predicts the next $k = 5000$ flips. To do this, we use the expectation of its final posterior to make k

[1] Code available at `https://github.com/lucy3/RRmemory`

random Bernoulli flips. The cost of prediction error is defined

$$C_E = \gamma \left| \frac{\hat{h}(C_F) - h}{k} \right|^2 \tag{2}$$

where h is the number of heads in the next k flips, $\hat{h}(C_F)$ is the prediction, when the model makes an effort payment of C_F, and γ is the scaling factor. A good calibration weight γ balances payment and error units so that our model allows us to solve the following optimization problem:

$$C^* = \arg\min_{C} C_E + C_F = \arg\min_{C_F} \gamma \left| \frac{\hat{h}(C_F) - h}{k} \right|^2 + C_F \tag{3}$$

In other words, we want our error to be significant enough for the model to pay some effort to reduce the noise, while also aware that paying too much does not increase the model performance proportionately. This way, we can verify if we can find the optimal solution under some environment.

The calibration factor γ we used for all models was 200.

Noise function

Functions 1 and 2 can be thought of as learners that try to retain the entire posterior distribution. Functions 3, 4, and 5 are ones that attempt to reconstruct the new prior using only some pieces or parameters of the posterior.

Function 0: Ideal learner. There is no noise added between the posterior after one observation and the prior for the next observation. This is equivalent to payment $\to \infty$ in the other noise functions.

Function 1: False memory. With a probability of $\frac{1}{C_F+1}$, where C_F is how much we pay and $C_F \geq 0$, the model makes an extra observation that is the opposite of the observation it will make in the next time step. In this equation, the probability that the model incorporate a false memory is inversely proportional to the effort cost. We add one to the cost in the denominator so that the equation will always be defined, even when cost is zero.

False memories have been studied extensively in psychology literature as an actual phenomenon that stems from the suggestibility of humans to false information (Schacter, 1999).

Function 2: Degradation. With a probability of $\frac{1}{C_F+1}$, where C_F is the effort payment and $C_F \geq 0$, the model returns the initial prior that it started with at the first time step. Otherwise, it returns the full posterior.

Function 3: Forgetting mean. We take 100 samples from our posterior after an observation and calculate the mean and variance. The mean has a sample of a Gaussian distribution added to it, where $\mu = 0$ and $\sigma = 1/(C_F + 1)$. We use the variance and noisified mean to create a logit-normal distribution which we use as our new prior. Since the standard deviation of this distribution and payment are inversely proportional, the severity to which the mean is distorted scales inversely with payment.

Function 4: Forgetting variance. Like in Function 3, we take 100 samples to reconstruct a new logit-normal distribution. The variance has the square of a sample of a Gaussian distribution added to it, where $\mu = 0$ and $\sigma = 1/(C_F + 1)$, while the mean is maintained.

Function 5: Sampling. With a probability of $C_F - \lfloor C_F \rfloor$, we take a sample size of $\lceil C_F + 2 \rceil$, otherwise we take a sample size of $\lfloor C_F + 2 \rfloor$. We use these samples to reconstruct the mean and variance of a new logit-normal distribution. The fewer samples we take, the more noise is involved.

Experiment 1: Results

Using the degradation noise function and the parameters we mentioned earlier, we can plot the final posterior distribution for belief of coin weight, learned by models that pay an effort cost of 0, 1, 4, and 10, as well as an ideal learner (Figure 1). Our model shows that our model's performance converges to a uniform distribution when we set our effort cost to 0, and to the ideal learner as the effort cost increases, with the most significant changes in our model in between (approximately at cost 4).

We can also look at the average performance of each noise function in making predictions (Figure 2). The x-axis is the payment made per day (C_F), while the y-axis is $C_F + C_E$. Intuitively, we are comparing our effort cost with a sum of our initial payment and "payment" for prediction mistakes.

As we can see, the optimal cost to pay ranges from $\approx$1, for false memory and forgetting variance, to $\approx$4, for degradation and forgetting mean. The optimal performance also varies, showing us that it is difficult to get low error with degradation and sampling no matter how much a learner pays.

The rate of that error changing with increasing payments also differs among noise functions. Forgetting the mean can be very detrimental, as seen by the high performance costs (the y-axis values) in the left side of that curve. Its contrast with forgetting variance is interesting because lower variance corresponds to lower confidence in a belief, while a drifted mean corresponds to a wrong average belief. Eventually, the slope of the upward trend with larger payments should converge to one, because as $C_E \to 0$, $C_F + C_E \to C_F$.

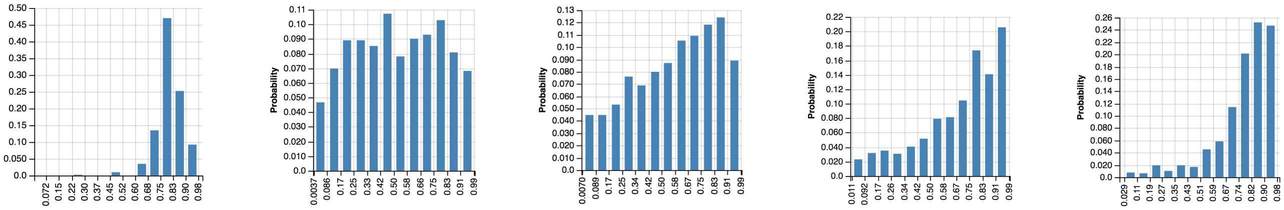
The main conclusion is that paying too much is not worth the further decrease in error, reducing the overall payoff.

Experiment 2: Methods

Overview

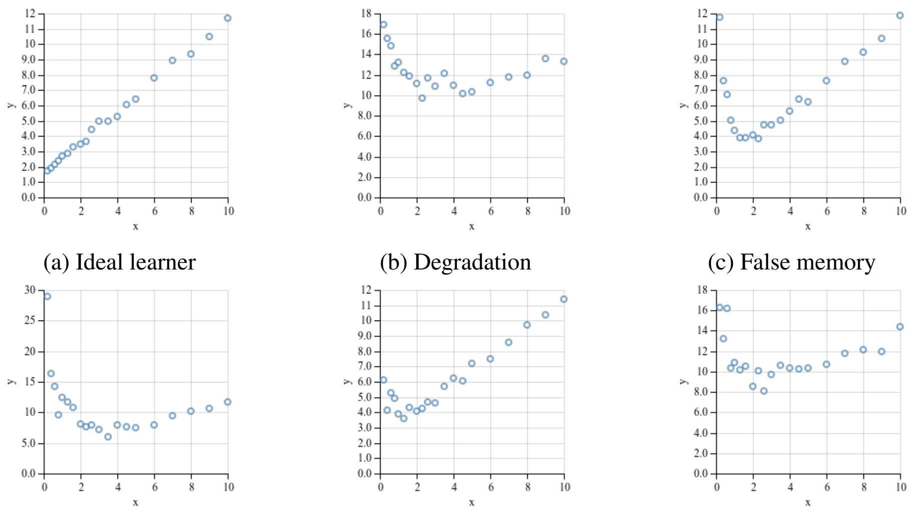
What if we didn't have to keep around the probabilities of a coin being a precise weight, but instead a more abstract idea, such as its category?

Expanding on the single-hierarchy case, we built a model based on a world where coins reside in two categories: heavy-weight coins, and light-weight coins. Heavy-weight coins have weights with a probability distribution represented by a logit-normal distribution with $\mu = 1$ and $\sigma = 1.5$, while light-weight coins are a logit-normal distribution with $\mu = -1$ and $\sigma = 1.5$. These distributions skew heavy-weight coins towards weights greater than 0.5, and light-weight coins towards those less than 0.5.



(a) Ideal learner (b) Payment = 0 (c) Payment = 1 (d) Payment = 4 (e) Payment = 10

Figure 1: Final posterior distributions.



(a) Ideal learner (b) Degradation (c) False memory

(d) Forgetting mean (e) Forgetting variance (f) Sampling

Figure 2: Performance over payments per noise function for the coin weight learner.

For the models in this section, the number of observations was 20, the actual coin weight was 0.8, and the calibration factor was 200, like in our single-hierarchy models.

Our framework for the model is the same as for Experiment 1, with the main difference in our observation step. At each time-step, we infer a new category belief by sampling from a prior category belief, sampling from the corresponding logit-normal coin weight distributions, and making an observation using that weight belief sample.

Noise function

Degradation. With a probability of $\frac{1}{C_F+1}$, the category belief returns to the uniform prior it started with at the first time step.

This function is analogous to the degradation noise function for coin weight. We can also try an analogous false memory noise function, but as seen in Figure 3, our model seems to become very confident in coin category very fast, and would be non-sensitive to the same rates of mistaken observations.

Evaluation function

We sample from our category belief, and using the result from that, sample from one of the two possible logit-normal coin weight distributions. We flip a coin of that sample weight to make a single prediction.

We repeat these actions 5000 times and use an error function like in Equation 2.

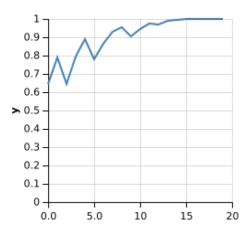
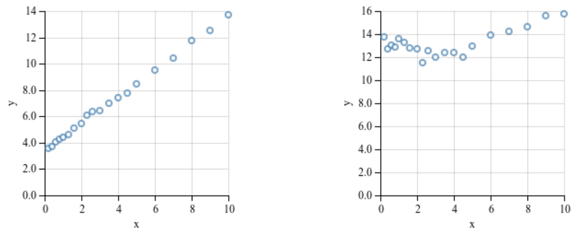


Figure 3: Ideal learner's probability of coin category = heavy.

Experiment 2: Results

The ideal learner becomes very confident in coin category quickly. For example, for a sequence of 20 observations consisting of (true, true, false, true, true, false, true, true, true, false, true, true, false, true, true, true, true, true, true, true) in that order, the probability of the coin being in the heavy category converges to > 0.95 after 13 days (Figure 3).

We can look at the average performance of our ideal cat-



(a) Ideal learner (b) Degradation

Figure 4: Performance over effort costs per noise function for the category learner.

egory learner and noisy learner in making predictions (Figure 4). The x-axis and y-axis are the same as Figure 2. The optimal cost to pay seems to be around 3, and the earlier part of the curve seems to be less sensitive to payment changes when compared to the curve for degradation noise in our single-hierarchy model. This suggests that a category learner seems to be more robust against noise.

However, the error for the ideal learner in this experiment is higher than the ideal learner for our single-hierarchy model. This makes intuitive sense because the information in a belief distribution about a weight of a coin is more precise than that in a distribution of coin category, and therefore has more predictive accuracy. However, our category learner does not do tremendously poorer than the coin weight learner, which means carrying less complex information may be more optimal if the number of "bits" we can afford to store between observations is limited.

Experiment 3: Methods

In the previous two experiments, we explored the trade-off between noise reduction and the model's prediction accuracy based on its coin parameter beliefs. We introduced an artificial noise function to represent flaws in human memory. However, with a hierarchical model, we should take into account the effects of information complexity on noise. Here, we explore the trade-off between the complexity of information at a knowledge level and our model's prediction accuracy in noiseless conditions.

For this experiment, we expand the ideal learner in Experiment 2 so that it can take on any number of upper-level categories, not just two (heavy-weight coins and light-weight coins). Increasing number of categories corresponds to increasing complexity at a knowledge level. For simple objects with one hidden parameter such as weighted coins, we only need to add a single upper hierarchical model with n categories, each one representing a prototype distribution of the coin weight. In our setting, we fixed the coin weight distribution of each category and update only the higher-level belief of what category the coin in question should belong to. We set the i^{th} category for the n-category model to have the following coin weight distribution:

$$w_{i,n} \sim \text{Beta}(i, n-1+1)$$

For each model, we begin with a discrete uniform distribution (with probability $\frac{1}{n}$ for each of the n coin types) as our prior belief of the coin's category. We make a number of observations of the coin flips and update the belief for each time step. Our goal is to use the final distribution to predict the coin weight by sampling 5000 coin weights p_{pred} from distribution and compare them against the the actual coin weight p_{actual}. In this experiment, we measure the root mean squared difference from the actual coin weight. This way, our measure will incorporate both accuracy and precision at an equal weight, since the expected deviation from the actual coin weight can be expanded into the following form:

$$\mathbf{E}(d_{\text{rms}}^2) = (\mathbf{E}(p_{\text{pred}}) - p_{\text{actual}})^2 + \text{Var}(p_{\text{pred}})$$

In the expression, the left term represents the deviation of the average coin weight according to some belief (the lower the more accurate), while the right term represents the deviation within the model itself (the lower the more precise). A good performing model will need to balance between both accuracy and precision.

Experiment 3: Results

We run the experiment on the actual coin weights $p_{\text{actual}} = 0.50$ and $p_{\text{actual}} = 0.85$, for number of categories n_c ranging from 1 to 30 and for the case of N observations ranging from 1 to 30. We plot the root mean square deviation d_{rms} in Figure 5. Increasing the number of categories lowers the deviation from the actual coin weight. Increasing the number of observations has a similar effect, to a lesser degree.

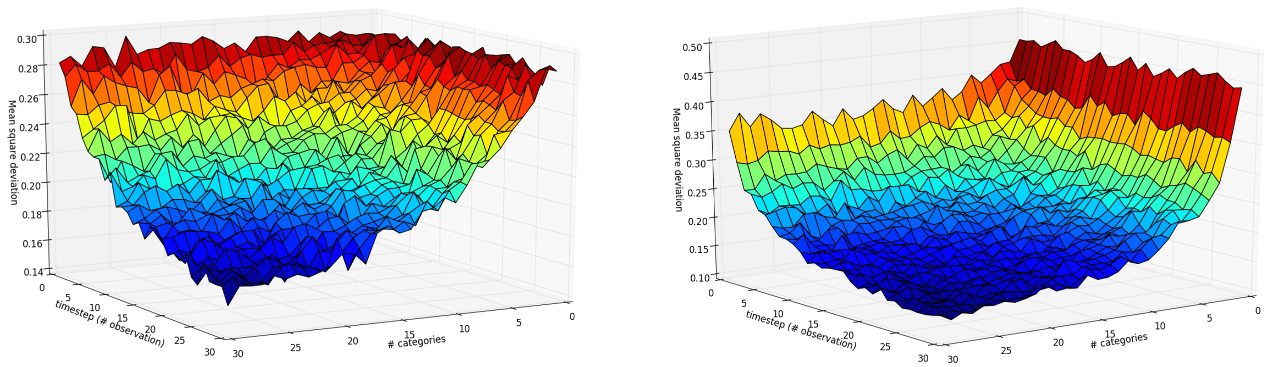
A notable feature in the result is, for coin weight $p_{\text{actual}} = 0.85$, our model starts learning as early as time step 3. For a fair coin, $p_{\text{actual}} = 0.5$, our model with 3 categories seems to learn very little. However, the belief for the categorical distribution shows noticable change from the original uniform distribution (Figures 6 & 7).

Another point that we can observe is that the mean square deviation from the actual coin weight always appear to be convex with the number of categories. That is, as the number of category (hierarchical complexity) increases, the performance of our model improves much slower, which draws much similarity to the results from the artificial cost functions proposed in experiments 1 and 2.

Next Steps

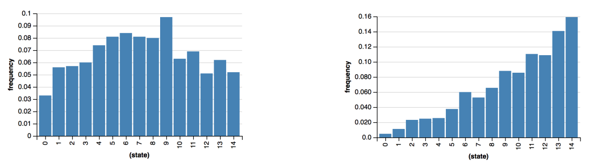
If we had all of the noise functions we proposed for our single-hierarchy model in play during learning and a limited amount of "money" to pay for all of them, it is clear that we would prioritize paying for a less noisy mean than other types of noise. However, to really formalize a solution for an optimization problem like this, we may be able to set it up as a constrained least squares problem.

We have seen that a learner is better at making predictions when the number of categories for information to fall into is increased (for example, warm and cold colors versus the seven colors of a rainbow). It would be interesting to define

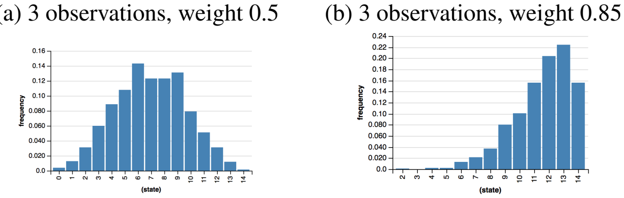


(a) coin weight 0.5 (b) coin weight 0.85

Figure 5: Deviation from the actual coin weight versus number of subcategories and number of observations.

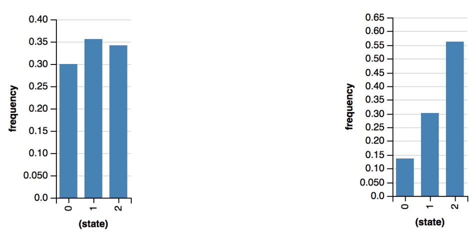


(a) 3 observations, weight 0.5 (b) 3 observations, weight 0.85

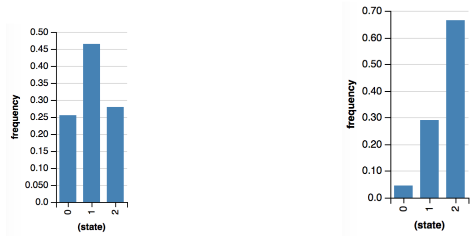


(c) 20 observations, weight 0.5 (d) 20 observations, weight 0.85

Figure 6: Final belief for 15 categories.



(a) 3 observations, weight 0.5 (b) 3 observations, weight 0.85



(c) 20 observations, weight 0.5 (d) 20 observations, weight 0.85

Figure 7: Final belief for 3 categories.

a relationship between cost and information complexity. For example, we would make it cheaper to keep around the abstract than the specific. In that scenario, it seems that it would be efficient to keep around higher-level information, but too high would result in too big of a drop in predictive accuracy.

Our models involve updating beliefs about a single coin, and it would be useful to expand beyond that. How can a model learn how a category affects lower-level information, even if that learning process is noisy? How could noisified upper-level beliefs generalize to other coins?

Finally, our payments have been constant for all observations. It would be useful to see how decreasing day-to-day payments between observations can still result in a "good enough" final posterior. It is hypothesized that once a model has learned enough, it does not have to be equally noise-adverse in the future.

Acknowledgments

The authors would like to express thanks to Michael Henry (MH) Tessler for guiding this project, writing framework code, and helping them navigate our ideas.

This project was initiated as part of Noah Goodman's class Computation and Cognition: The Probabilistic Approach (CS428/Psych204).

References

Anderson, J. R., & Schooler, L. J. (1991). Reflections of the Environment in Memory. *Psychological Science*, *2*(6), 396–408. doi: 10.1111/j.1467-9280.1991.tb00174.x

Chun, M. M., & Turk-Browne, N. B. (2007). Interactions between attention and memory. *Current Opinion in Neurobiology*, *17*(2), 177–184. doi: 10.1016/j.conb.2007.03.005

Goodman, N. D., & Stuhlmüller, A. (2014). *The Design and Implementation of Probabilistic Programming Languages.* `http://dippl.org`. (Accessed: 2016-11-18)

Goodman, N. D., & Tenenbaum, J. B. (n.d.). *Probabilistic Models of Cognition.* `https://probmods.org`. (Accessed: 2016-11-18)

Klosterman, C. (2003). *Sex, drugs, and coco puffs: A low culture manifesto*.

Murre, J. M. J., & Dros, J. (2015). Replication and analysis of Ebbinghaus' forgetting curve. *PLoS ONE*, *10*(7), 1–23. doi: 10.1371/journal.pone.0120644

Rosch, E., Mervis, C., Gray, W., Johnson, D., & Boyes-Brahem, P. (1976). Basic objects in natural categories. *Cognitive Psychology*, *8*, 382–439.

Schacter, D. L. (1999). The seven sins of memory: insights from psychology and cognitive neuroscience. *American psychologist*, *54*(3), 182. doi: 10.1037/0003-066X.54.3.182

Simon, H. (1957). *Models of Man: Social and Rational-Mathematical Essays on Rational Human Behavior in a Social Setting*. Wiley-first edition.

Spiro, R. J. (1980). Accommodative reconstruction in prose recall. *Journal of Verbal Learning and Verbal Behavior*, *19*(1), 84–95. doi: 10.1016/S0022-5371(80)90548-4

Vul, E., Goodman, N., Griffiths, T. L., & Tenenbaum, J. B. (2014). One and done? Optimal decisions from very few samples. *Cognitive Science*, *38*(4), 599–637. doi: 10.1111/cogs.12101

Xu, J., & Griffiths, T. L. (2010). A rational analysis of the effects of memory biases on serial reproduction. *Cognitive Psychology*, *60*(2), 107–126. Retrieved from `http://dx.doi.org/10.1016/j.cogpsych.2009.09.002` doi: 10.1016/j.cogpsych.2009.09.002