Reward Generalization and Reward-Based Hierarchy Discovery for Planning

Agni Kumar
Computer Science and Engineering
Massachusetts Institute of Technology
Cambridge, MA 02139
agnik@mit.edu

Samyukta Yagati
Computer Science and Engineering
Massachusetts Institute of Technology
Cambridge, MA 02139
samyu@mit.edu

Abstract

Humans have the unique ability to break apart their intents into high-level objectives that allow them to accomplish tasks. When applied to computationally intelligent agents, this methodology (known as hierarchical planning) could allow for models with advanced planning abilities. This hierarchy can be modeled from a Bayesian viewpoint by assuming a generative process for the structure of a particular environment. Existing work on this problem includes the development of a computational framework for acquiring hierarchical representations under a set of simplified assumptions on the hierarchy: that is, modeling how people create clusters of states in their mental representations of reward-free environments in order to facilitate planning. We contribute a Bayesian cognitive model of hierarchical discovery which combines knowledge of clustering and rewards to predict cluster formation, and compare the model to human data. We analyzed situations with both static and dynamic reward mechanisms. We found that (1) humans generalize information about rewards to high-level clusters, and use information about rewards to create clusters, and (2) the reward generalization and reward-based cluster formation can be predicted by our proposed model.

1 Introduction

1.1 Example

Suppose Jane, sitting in her Boston apartment, gets a notification from her boss that there is an important business meeting happening in Paris next week. How will Jane go about planning this trip? Rather than go about goal achievement through a series of detailed actions ("get out of bed", "go to desk", "open laptop", "search flights to Paris", etc.), Jane starts with actions that will take her generally in the direction of her broader goal and increases granularity of planning as she approaches her goal. She might first look for flights to Paris, and then at how to rent a car in the Paris airport, and finally which combination of streets and turns she should take to drive from the airport to the office where the meeting is located. This type of planning requires a hierarchical representation of the world.

1.2 Background

A key area where psychology and neuroscience combine is the formal understanding of human behavior in relation to assigned actions. Specifically, what is the planning and methodology employed by human agents when faced with accomplishing some task? This is especially interesting in light of the unique ability of humans and animals to adapt to new environments. Previous literature on animal learning suggests that this flexibility stems from a hierarchical representation of goals that allows for

complex tasks to be broken up into low-level subroutines that can be extended across a variety of contexts.

This process, known as "chunking," occurs when actions are stitched together into temporally extended action sequences that achieve distant goals. Chunking is often the result of the transfer of learning from a goal-directed system to a habitual system, which executes actions in a stereotyped way. From a computational standpoint, such a hierarchical representation allows for agents to quickly execute actions in an open loop, reuse familiar action sequences whenever a known problem is encountered, learn faster by tweaking established action sequences to solve problems reminiscent of those seen previously, and plan over extended time horizons, because agents do not need to be concerned with the minuscule tasks associated with goal achievement: for example, the goal of going to the store being broken down into leaving the house, walking, entering the store as opposed to getting up from the bed, moving the left foot forward, right foot forward, left foot forward, etc.

Hierarchical reinforcement learning (HRL) has become the prevailing framework for representing hierarchical learning and planning. Within research on modeling of HRL, several ideas have presented around potential methods of model construction. Simsek et al. presents a method based on "betweenness", a graph centrality concept measuring the shortest paths that go through every vertex of a graph[20]. Through the construction of an interaction graph representing potential state transitions, a metric is produced indicative of the cost of each individual path. These weights are then used to divide and group clusters of low-level tasks. Meanwhile, Van Dijk et al. defined subgoals as states in which there is significant change in the amount of relevant goal information and used this delta value to construct a potential hierarchical structure[19]. In addition to these studies exploring the constructions of hierarchies, work by Solway et al. looked to evaluate these models to identify the best HRL for a particular behavior domain. It was proposed that the optimal hierarchy is one that facilitates adaptive behavior when presented with a new problem; this was formerly defined as a Bayesian model selection[1].

1.3 Objective

People spontaneously organize their environment into clusters of states that constrain planning. Such hierarchical planning is more efficient in time and memory than "flat" planning, which is consistent with people's limited working memory capacity[5]. Solway et al. provide a formal definition of an "optimal" hierarchy, but they do not specify how the brain might discover it[1]. We hypothesize that an optimal hierarchy depends on the structure of environment, including both graph structure and the distribution of observable features of the environment, specifically rewards. In this work, we propose a Bayesian cognitive model of hierarchy discovery based on topological structure reward distribution and show that it is consistent with human behavior.

2 Model

We assumed that agents represent their environment as a graph, where nodes are states in the environment and edges are transitions between states. The states and transitions may be abstract, or as concrete as subway stations and the train lines traveling between them. The parts of the model described in Sections 2.1 and 2.2 were previously implemented. We implemented the addition to the model of reward generalization and reward-based clustering, described in Section 2.3.

2.1 Setup

We assume a 2-layer hierarchy. $G = (V, E)$ is the observable graph, where V is the set of vertices and $E \subset \{V \times V\}$.

$H = (V', E', c, b, p', p, q)$ is the non-observable hierarchy (i.e. the latent structure), where:

- V' is the set of high-level vertices (high-level states, or clusters)
- $E' : \{V' \times V'\}$ is the set of high-level edges
- $c : V \to V'$ are the cluster assignments
- $b : E' \to E$ are the bridges
- $p' \in [0, 1]$ is the density of the high-level graph

- $p \in [0, 1]$ is the within-cluster density of G
- $q \in [0, 1]$ is the across-cluster density penalty of G

Both G and H are unweighted, undirected graphs. Informally, H consists of clusters, where each low-level node in G belongs to exactly one cluster, and bridges, or high level edges, that connect these clusters. Bridges can exist between clusters k and k' only if there is an edge between some $v, v' \in V$ such that $v \in k$ and $v' \in k'$. That is, each high-level edge in H has a corresponding low-level edge in G.

2.2 Model Structure

The existing algorithm discovers optimal hierarchies given the following constraints:

1. Small clusters
2. Dense connectivity within clusters
3. Sparse connectivity across clusters

However, we do not want clusters to be too small (in the extreme, each node is its own cluster, which renders the hierarchy useless). Additionally, while we want sparse connectivity across clusters, we want to maintain "bridges," or single-edge connections, across clusters in order to preserve properties of the underlying graphs. The following generative model generates hierarchies that respect these constraints.

$$
\begin{aligned}
c &\sim CRP(\alpha) && \text{cluster assignments} \\
p' &\sim Beta(1,1) && \text{H graph density} \\
Pr[(k,l) \in E'] &= p' && \text{H graph edges} \\
Pr[b_{k,l} = (i,j) \mid (k,l) \in E', c_i = k, c_j = l] &= \frac{1}{n_k n_l} && \text{bridges} \\
p &\sim Beta(1,1) && \text{within-cluster density} \\
q &\sim Beta(1,1) && \text{cross-cluster density penalty} \\
Pr[(i,j) \in E \mid c_i = c_j] &= p && \text{within-cluster edges} \\
Pr[(i,j) \in E \mid c_i \neq c_j] &= pq && \text{cross-cluster edges} \\
Pr[(i,j) \in E \mid b_{c_i,c_j} = (i,j)] &= 1 && \text{bridge edges}
\end{aligned}
$$

We use the discrete-time stochastic CRP (Chinese Restaurant Process) as a prior for clusters. The discovery of hierarchies can be accomplished by inverting the generative model to obtain the posterior probability of hierarchy H:

$$
\begin{aligned}
P(H|G) &\propto P(G|H)P(H) \\
&= P(E|c,b,p,q)P(p)P(q)P(b|E',c)P(E'|p')P(p')P(c)
\end{aligned}
$$

2.3 Rewards

The focus of this paper is the addition of reward learning to the model described in Sections 2.1 and 2.2. In the context of the graph G, rewards can be interpreted as observable features of vertices. Because people often cluster based on observable features, it is reasonable to model clusters induced by rewards[3]. Furthermore, we assume that each state delivers a randomly determined reward, and that the agent's goal is to maximize total reward[7].

Since we hypothesize that clusters induce rewards, we model each cluster as having an average reward. Each node in that cluster has an average reward drawn from a distribution centered around the average cluster reward. Finally, each observed reward is drawn from a distribution centered around the average reward of that node.

More formally, we can incorporate rewards into the generative model with latent variables θ and μ, and the observed variable r:

$$\begin{aligned} \theta_k &\sim \mathcal{N}(\bar{\theta}, 100) && \text{average cluster rewards} \\ \mu_i &\sim \mathcal{N}(\theta_{c_i}, 100) && \text{average state rewards} \\ r_{i,t} &\sim \mathcal{N}(\mu_i, \sigma_r^2) && \text{rewards} \end{aligned}$$

where $k \in V'$, $i \in V$, θ_k is the average reward for cluster k, μ_i is the average reward for node i (whose cluster is denoted c_i), and $r_{i,t}$ is the reward observed for node i at time t.

Static rewards. To simplify inference, we first assume that rewards are constant, or the variance of the reward $\sigma_r^2 = 0$. This is referred to as the static rewards case in the rest of the paper.

Dynamic rewards. In the case of dynamic rewards, we assume that rewards can change between observations with some fixed probability.

3 Experiments

We conducted two experiments to test our hypothesis about human behavior, and to understand how well it could be predicted by our proposed model. In particular, we studied to what degree clusters drive inferences about rewards, and to what degree rewards drive the formation of clusters. For each experiment, we collected human data and compared it to the predictions of the model.

For the following experiments, we used a constant setting for our hyperparameter values σ_θ^2, $\bar{\theta}$, σ_μ^2, and σ_r^2. $\bar{\theta}$ was set to the desired mean of cluster rewards across the graph, which was 15 in all experiments. We set $\sigma_\theta^2 = 10$ so that cluster means would be allowed to vary but typically stay above zero. We set $\sigma_\mu^2 = 10$ as well. This is the variance of the mean reward of each individual node from the mean reward of the cluster it belongs to. Finally, we set $\sigma_r^2 = 5$. This is the variance of observed rewards at an individual node from the mean reward μ at that node. While it would ideally be zero in order to model constant rewards, making it very small or zero would result in Metropolis-within-Gibbs frequently guessing impossible hierarchies. We wanted to keep this value as small as possible so that deviations of the observed reward from $\bar{\theta}$ would be explained primarily by θ and μ. The value of 5 satisfied both constraints.

3.1 Experiment 1: Clusters induce rewards

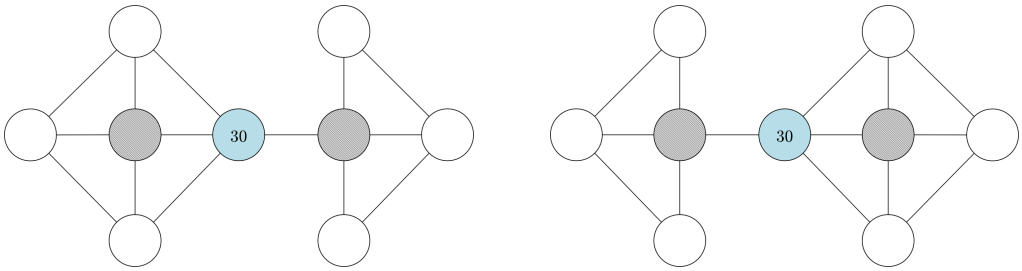
The goal of the first experiment was to understand how clusters induce rewards. We tested whether graph structure drives cluster formations and whether people generalize a reward observed at one node to the cluster that the node belongs to.

3.1.1 Setup

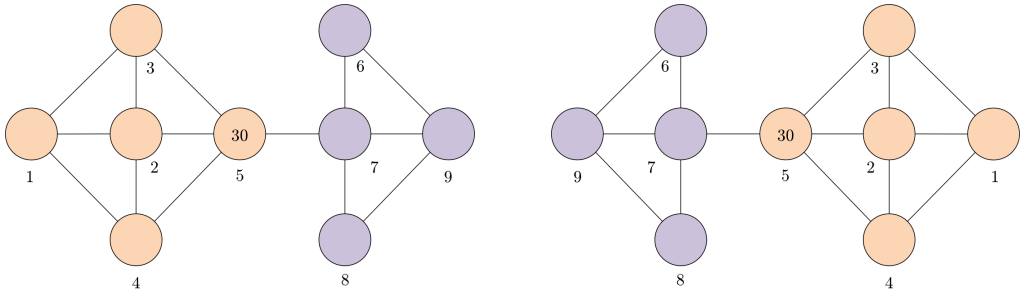
This experiment was conducted by asking 32 human subjects to choose a node to visit next as specified in the following scenario. Participants were randomly presented with one of the two graphs below, to ensure that bias of handedness or graph structure was not introduced. We predicted that participants would choose the node adjacent to the labeled one that was located in the larger cluster, i.e. the gray node to the left of the blue one in the first case, and the gray node to the right of the blue one in the second case.

You work in a large gold mine that is composed of multiple individual mines and tunnels. The layout of the mines is shown in the diagram below (each circle represents a mine, and each line represents a tunnel). You are paid daily, and are paid $10 per gram of gold you found that day. You dig in exactly one mine per day, and record the amount of gold (in grams) that mine yielded that day.

Over the last few months, you have discovered that, on average, each mine yields about 15 grams of gold per day. Yesterday, you dug in the blue mine in the diagram below, and got 30 grams of gold. Which of the two shaded mines will you dig in today? Please circle the mine you choose.



We expected most participants to automatically identify the following clusters, with nodes colored in peach and lavender to denote the different clusters, and make a decision about which mine to select with these clusters in mind. It was hypothesized that participants would select a peach-colored node as opposed to a lavender one, since the node with label 30, a fairly-larger-than-average reward, is in the peach-colored cluster.



3.1.2 Inference

We approximated Bayesian inference over H using Metropolis-within-Gibbs sampling[8], which updates each component of H by sampling from its posterior, conditioning on all other components in a single Metropolis-Hastings step. We employed a Gaussian random walk as the proposal distribution for continuous components, and the conditional Chinese restaurant process (CRP) prior as the proposal distribution for cluster assignments[9]. Essentially, the approach can be interpreted as as stochastic hill climbing with respect to a utility function defined by the posterior.

3.1.3 Results

The top three clusterings outputted by the model outputted by the model are shown in *Figure 1*. The results for participants, as well as those for the static rewards model, are visualized in Table **??** and in *Figure 2*. There were 32 participants total in each of the human and simulated groups.

	Picked node 2	Picked node 7	p-value
Human	24	8	0.0011
Simulation	21	11	0.0251

Table 1: p-values for Experiment 1.

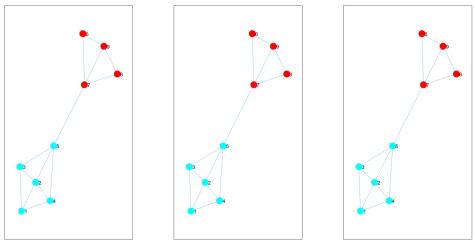


Figure 1: Clusterings identified by the static rewards model. All top three results were the same, indicating that the model identified the colored groupings with high confidence.

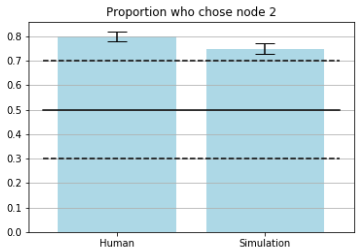


Figure 2: Proportion of human and simulated subjects, out of 32 total, who chose to visit node 2 next. The solid black line indicates the mean and the dotted black lines indicate the 10^{th} and 90^{th} percentiles.

The listed p-values in the table above were calculated via a right-tailed binomial test, where the null was assumed to be a binomial distribution over choosing left or right gray node. The significance level was taken to be 0.05, and both the human experimental results and modeling results were statistically significant.

3.2 Experiment 2: Rewards induce clusters

In the second experiment, the goal was to determine whether rewards induce clusters. We predicted that nodes with the same reward that were positioned adjacent to each other would be clustered together, even if the structure of the graph alone would not induce clusters. Recall that Solway et. al showed that people prefer paths that cross the fewest hierarchy boundaries[1]. Therefore, between two otherwise identical paths from node A to node B, the only reason to prefer one over the other would be because it crosses fewer hierarchy boundaries. One possible counterargument to this is that people pick the path with higher rewards. However, in our setup (see section 3.2.1), rewards are given only in the goal state, not cumulatively over the path taken. Additionally, the magnitude of rewards was varied between trials. Therefore, it is unlikely that people would favor a path because nodes along that path had higher rewards.

3.2.1 Setup

This experiment was conducted on the web using Amazon Mechanical Turk[1]. *Figure 3* shows the diagram presented to participants. As in Experiment 1, participants were randomly given either the configuration shown in *Figure 3* or the horizontally-flipped version of the same graph in order to control for potential left-right asymmetry. Similarly to experiment 1, participants were given the following context about the task:

[1]The human data for this experiment was collected by Momchil Tomov, a graduate student at the Department of Psychology and Center for Brain Science at Harvard University.

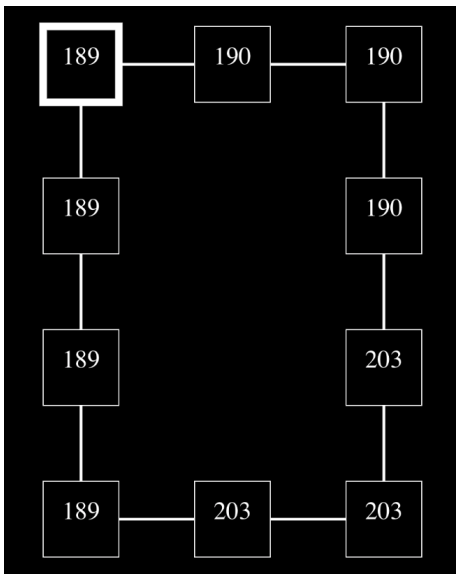


Figure 3: The diagram presented to participants.

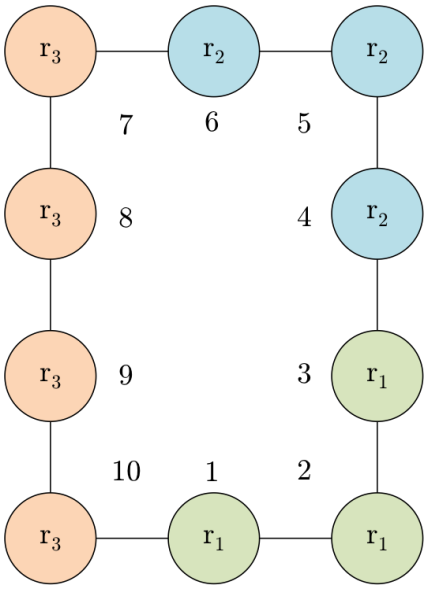


Figure 4: The nodes are numbered for reference. The expected induced clusters are indicated by color.

Imagine you are a miner working in a network of gold mines connected by tunnels. Every mine yields a certain amount of gold (points) each day. On each day, your job is to navigate from a starting mine to a target mine and collect the points from the target mine.

On some days, you will be free to choose any mine you like. On those days, you should try to pick the mine that yields the most points.

On other days, only one mine will be available. The points of that mine will be written in green and the other mines will be grayed out. On those days, you should navigate to the available mine.

The points of each mine will be written on it. The current mine will be highlighted with a thick border. You can navigate between mines using the arrow keys (up, down, left, right). Once you reach the target mine, press the space key to collect the points and start the next day. There will be 100 days (trials) in the experiment.

We will refer to the first case (where participants are free to navigate to any mine) as *free-choice* and the second case (where participants navigate to a specified mine) as *fixed-choice*. We note that participants received a monetary reward (one cent per point earned) for each trial to discourage random responses.

At each trial, reward values were changed with probability 0.2. New rewards were drawn uniformly at random from the interval [0, 300]. However, the grouping of rewards remained the same across trials: nodes 1, 2, and 3 (see *Figure 4*) always had one reward value; nodes 4, 5, and 6 had a different reward value; and nodes 7, 8, 9, and 10 had a third reward value. The first 99 trials allowed the participant to develop a hierarchy of clusters. The final trial, which acted as the test trial, asked participants to navigate from node 6 to node 1 (*Figure 4*). Assuming that rewards induced the clusters shown in *Figure 4*, we predicted that more participants would take the path through node 5, which crosses only one cluster boundary, than through node 7, which crosses two cluster boundaries, in keeping with existing findings[1].

	Via node 5	Via node 7	p-value
Human	56	39	0.0321
Static Rewards Simulation	64	31	0.0002
Dynamic Rewards Simulation	54	41	0.0753

Table 2: p-values for Experiment 2.

3.2.2 Inference

We modeled the fixed-choice case, with the assumption that the tasks in all 100 trials were all the same as the 100th trial presented to participants (the test trial). First, we assumed static rewards, where the rewards remained constant across all trials. Next, we assumed dynamic rewards, where rewards changed for each trial as described in section 3.1.1. In contrast to Experiment 1, where the participant picks a single node the model predicts that node, Experiment 2 is concerned with the second node of the full path the participant (or simulated participant) chose to take from the start node to the goal node. Therefore, in order to compare the model to human data, we used a variant of breadth-first search, hereafter referred to as hierarchical BFS, to predict a path from the start node (node 6) to the goal (node 1).

Static rewards. For each subject, we sampled from the posterior using Metropolis-within-Gibbs sampling and chose the most probable hierarchy (i.e. the hierarchy with the highest posterior probability), as in Experiment 1. Then, we used hierarchical BFS to first find a path between clusters and then between the nodes within the clusters.

Dynamic rewards. For dynamic rewards, we used online inference. For each simulated participant, we allowed the sampling for each trial to progress for only 10 steps. Then, we saved the hierarchy, and added information about the modified rewards. Next, we allowed sampling to progress again, starting from the saved hierarchy. The saved hierarchy was simply the final hierarchy generated by the sampling process; although we used the maximum a posteriori hierarchy to perform the final inference step, we did not track it throughout the inference process. As in the human experiment, rewards were redrawn at the beginning of each trial with probability 0.2, and the rewards were always equal within clusters. The simulated rewards were scaled to be float values in the interval [0, 30], rather than the integers in the interval [0, 300] used for the human experiment. This inference method simulates how human participants might learn cumulatively over the course of many trials. We assumed, for the purpose of this experiment, that people keep only one hierarchy in mind at a time, rather than updating multiple hierarchies in parallel. We also modified the log posterior to penalize disconnected clusters, because such clusters became much more common under this type of inference.

3.2.3 Results

The results of Experiment 2 are summarized in Table 2, *Figure 5*, and *Figure 6*. The table shows the number of participants who chose the path through node 5 and through node 7. In each group (human, and each simulated group), there were 95 total participants. Under the null hypothesis, the distribution of choices follows a binomial distribution with $n = 95$, the number of subjects, and $p = 0.5$. Given n experiments in which each has a yes-no outcome (in our case, a choice of node 5 or node 7) with a probability p of success and probability $1 - p$ of failure, a binomial distribution describes the number of successes (a choice of node 5, in our experiment). Therefore, it is an appropriate choice of null hypothesis in this context.

As in Experiment 1, *p*-values were computed with a right-tail binomial test. The results of the human experiment and static rewards modeling were statistically significant, since the respective *p*-values were less than the significance level, 0.05 (see Table 2). Furthermore, as shown in *Figure 6*, the results of the human experiment are in the 90^{th} percentile of a normal distribution centered around 0.5, the expected proportion given the null hypothesis.

Static rewards. We used 1000 iterations of Metropolis-within-Gibbs to generate each sample, with a burnin and lag of 1 each. The simulation under static rewards favors paths through node 5 to a level that is statistically significant with a significance level of 0.05. Moreover, since its purpose is to model human behavior, this result is meaningful in light of the human data being statistically significant as well ($0.0321 < 0.05$).

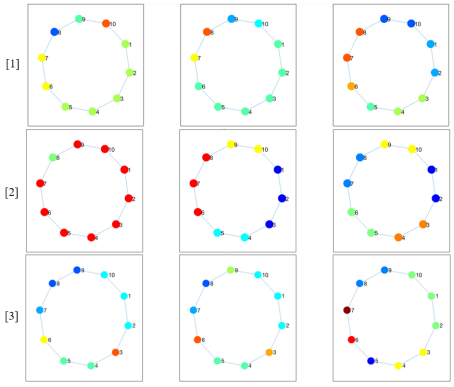


Figure 5: Clusterings identified by [1] the static rewards model, [2] the static rewards model with cluster formation between disconnected components penalized, and [2] the dynamic rewards model.

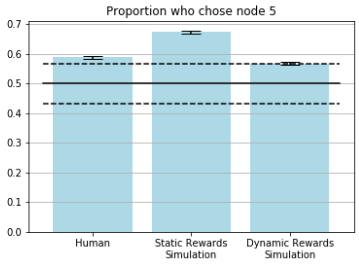


Figure 6: Proportion of human and simulated subjects, out of 95 total, whose chosen path's second node was node 5. The solid black line indicates the expected proportion given the null hypothesis and the dotted black lines indicate the 10^{th} and 90^{th} percentiles.

Dynamic rewards. For each simulated subject, in order to mimic the human trials, we ran 100 trials, each with 10 iterations of Metropolis-within-Gibbs to sample from the posterior. Burnin and lag were set to 1. The dynamic rewards model did not produce results that were statistically significant; we hope to develop the model further in the future. However, the online inference method (dynamic rewards modeling) appears to have modeled human data better than modeling for static rewards, even though the group of simulated participants under dynamic rewards modeling is farther from the hypothesis (i.e. less frequently chooses the route that traverses the fewest cluster boundaries) than the group simulated under static rewards modeling. 56 human participants and 54 simulated participants under dynamic rewards modeling chose to go through node 5 (a 3.4% difference), compared to 64 simulated participants under static rewards modeling (an 18.5% difference).

4 Conclusion

We have shown, through collection of relevant human data, that an optimal hierarchy depends on the environment structure. In particular, the optimal hierarchy depends not only on graph structure, but also on observable characteristics of the environment, i.e. the distribution of rewards. We built hierarchical Bayesian models to understand how clusters induce static rewards (Experiment 1) and how both static and dynamic rewards induce clusters (Experiment 2), and found that most results were statistically significant in terms of how closely our models captured human actions.

5 Acknowledgements

We would like to express our sincere gratitude to the teaching staff of *Computational Cognitive Science* for supporting our project and organizing the class. We would also like to extend a big thank you to Momchil Tomov, our project TA!

6 Division of Work

We both contributed to modeling for Experiment 1, and contributed equally to the writing of this report. Modeling for Experiment 2 was done by Samyu, and the human data for Experiment 1 was collected by Agni.

References

[1] Alec Solway, Carlos Diuk, Natalia Córdova, Debbie Yee, Andrew G. Barto, Yael Niv, and Matthew M. Botvinick. Optimal behavioral hierarchy. *PLOS Computational Biology*, 10(8):1–10, 08 2014.

[2] Anna C. Schapiro, Timothy T. Rogers, Natalia I. Cordova, Nicholas B. Turk-Browne, and Matthew M. Botvinick. Neural representations of events arise from temporal community structure. *Nature Neuroscience*, 16(4):486–492, Apr 2013.

[3] Jan Balaguer, Hugo Spiers, Demis Hassabis, and Christopher Summerfield. Neural mechanisms of hierarchical planning in a virtual subway network. *Neuron*, 90(4):893–903, 2016.

[4] Juan A Fernández and Javier González. *Multi-hierarchical representation of large-scale space: Applications to mobile robots*, volume 24. Springer Science & Business Media, 2013.

[5] George A Miller. The magic number seven plus or minus two: Some limits on our capacity for processing information. *Psychological review*, 63:91–97, 1956.

[6] Samuel J Gershman and David M Blei. A tutorial on bayesian nonparametric models. *Journal of Mathematical Psychology*, 56(1):1–12, 2012.

[7] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[8] Gareth O Roberts and Jeffrey S Rosenthal. Examples of adaptive mcmc. *Journal of Computational and Graphical Statistics*, 18(2):349–367, 2009.

[9] Radford M Neal. Markov chain sampling methods for dirichlet process mixture models. *Journal of computational and graphical statistics*, 9(2):249–265, 2000.

[10] Christian P Robert. Casella: Monte carlo statistical methods. *Springerverlag, New York*, 3, 2004.

[11] Arnaud Doucet, Nando De Freitas, and Neil Gordon. An introduction to sequential monte carlo methods. In *Sequential Monte Carlo methods in practice*, pages 3–14. Springer, 2001.

[12] Pratiksha Thaker, Joshua B Tenenbaum, and Samuel J Gershman. Online learning of symbolic concepts. *Journal of Mathematical Psychology*, 77:10–20, 2017.

[13] Nicolas Chopin. A sequential particle filter method for static models. *Biometrika*, 89(3):539–552, 2002.

[14] Thomas H Cormen, Charles E Leiserson, Ronald L Rivest, and Clifford Stein. *Introduction to algorithms*. MIT press, 2009.

[15] Kevin P Murphy. Active learning of causal bayes net structure. 2001.

[16] Simon Tong and Daphne Koller. Active learning for structure in bayesian networks. In *International joint conference on artificial intelligence*, volume 17, pages 863–869. Citeseer, 2001.

[17] Chandra Nair, Balaji Prabhakar, and Devavrat Shah. On entropy for mixtures of discrete and continuous variables. *arXiv preprint cs/0607075*, 2006.

[18] Mark Steyvers, Joshua B Tenenbaum, Eric-Jan Wagenmakers, and Ben Blum. Inferring causal networks from observations and interventions. *Cognitive science*, 27(3):453–489, 2003.

[19] Van Dijk SG, Polani D, Nehaniv CL. Hierarchical behaviours: getting the most bang for your bit. *Advances in Artificial Life: Darwin Meets von Neumann*, 342-349, 2011.

[20] Simsek O, Bart AG. Skill characterization based on betweenness *Advances in Neural Information Processing Systems*, 1497-1504, 2008.

[21] Relevant code can be found at https://github.com/agnikumar/chunking.