Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI-16)

Modeling Human Understanding of Complex Intentional Action with a Bayesian Nonparametric Subgoal Model

Ryo Nakahashi
Computer Science and
Artificial Intelligence Laboratory
Massachusetts Institute of Technology, USA
Sony Corporation, JAPAN
Ryo.Nakahashi@jp.sony.com

Chris L. Baker and **Joshua B. Tenenbaum**
Department of Brain and Cognitive Sciences
Massachusetts Institute of Technology, USA
{clbaker,jbt}@mit.edu

Abstract

Most human behaviors consist of multiple parts, steps, or subtasks. These structures guide our action planning and execution, but when we observe others, the latent structure of their actions is typically unobservable, and must be inferred in order to learn new skills by demonstration, or to assist others in completing their tasks. For example, an assistant who has learned the subgoal structure of a colleague's task can more rapidly recognize and support their actions as they unfold. Here we model how humans infer subgoals from observations of complex action sequences using a nonparametric Bayesian model, which assumes that observed actions are generated by approximately rational planning over unknown subgoal sequences. We test this model with a behavioral experiment in which humans observed different series of goal-directed actions, and inferred both the number and composition of the subgoal sequences associated with each goal. The Bayesian model predicts human subgoal inferences with high accuracy, and significantly better than several alternative models and straightforward heuristics. Motivated by this result, we simulate how learning and inference of subgoals can improve performance in an artificial user assistance task. The Bayesian model learns the correct subgoals from fewer observations, and better assists users by more rapidly and accurately inferring the goal of their actions than alternative approaches.

Introduction

Human behavior is hierarchically structured. Even simple actions – checking email, for example – consist of many steps, which span levels of description and complexity: *moving* fingers, arms, eyes; *choosing* whether to use the mouse or keyboard; *searching* for the email app among the others that are open, etc. Human behavior is also efficient: we attempt to perform each part of each action as swiftly and successfully as we can, at the least possible cost.

Classical models of how humans understand the actions of others (originating from the plan recognition literature (Schank and Abelson 1977; Kautz and Allen 1986; Charniak and Goldman 1993; Bui, Venkatesh, and West 2002) seek to leverage this hierarchical structure by building in prior knowledge of others' high-level actions, tasks, and goals. Behavioral evidence has shown that adults and even infants can learn simple hierarchical action structures from data, segmenting novel sequences along statistical action boundaries (Baldwin et al. 2008). These abilities can be captured with a nonparametric action segmentation model (Buchsbaum et al. 2015), which learns lists of actions that occur in sequence, and potentially cause observable effects.

However, purely statistical learning from data leaves implicit the intrinsic efficiency of intentional actions. For adults and infants, the assumption that others' actions are rational functions of their beliefs, desires, and goals is fundamental (Dennett 1987; Gergely et al. 1995). Rather than simply memorizing repeated action sequences, people infer goals of complexity sufficient to rationalize these actions (Schachner and Carey 2013). A simple formalization of people's theory of rational, goal-directed action can be given in terms of approximately rational planning in Markov decision processes (MDPs), and human goal inferences can be accurately predicted using Bayesian inference over models of MDP planning (or inverse planning) (Baker, Saxe, and Tenenbaum 2009). This approach is closely related to recent plan recognition algorithms developed in the AI literature (Ramírez and Geffner 2010, e.g.).

In this paper, we integrate hierarchically structured actions into the inverse planning framework using hierarchical MDPs (Sutton, Precup, and Singh 1999). We consider scenarios in which agents pursue sequences of subgoals enroute to various destinations. Each destination can have multiple subgoal sequences, generated by a nonparametric Bayesian model. Together, a destination and subgoal sequence induce a hierarchical MDP, and agents' actions are assumed to be generated by approximately rational planning in this MDP.

Representing agents' subgoal-based plans allows segmentation of behavior into extended chunks, separated by sparse subgoal boundaries, and can achieve greater generalization than purely statistical models by naturally capturing the context-sensitivity of rational action sequences, rather than having to learn new sequences for each context. Further, the model predicts deviations from rationality (with respect to a single subgoal) at subgoal boundaries – a strong cue to hierarchical structure. This inductive bias should enable efficient learning of subgoal structure from small numbers of examples.

We present two experiments to test our model. The first is a behavioral experiment, in which human participants inferred the subgoal structure underlying series of action se-

quences. The second experiment is a simulation to show that the model is useful in an artificial user support task. The model first learns the subgoal structure of the task from a small number of observations. Then, the model infers a user's destination and subgoals from a partial action sequence, and attempts to assist the user to achieve a subset of the remaining subgoals. For each experiment, we compare the performance of our model with that of several natural alternatives.

Computational Model

Fig. 1 represents the structure of our model in terms of separate graphical models for the hierarchical MDP and nonparametric subgoal models. These graphical models specify the structure of the joint distributions over actions, state sequences, and subgoal sequences. Our model represents the situational context in terms of a finite state space $\mathcal{S}$. The variable $\boldsymbol{s}$ denotes a state sequence of length T, such that $s_t \in \mathcal{S}$ is the tth state in the sequence. The variable $\boldsymbol{g}$ represents a sequence of M subgoals. We denote the mth subgoal of $\boldsymbol{g}$ as $g_m \in \mathcal{S}$. For convenience, we assume that $s_T = g_M = d$ is the destination. We denote the set of actions as $\mathcal{A}$, and the action executed in s_t as $a_t \in \mathcal{A}$ (see Fig. 1(a)).

The remainder of this section will first define the hierarchical MDP induced by a given destination and subgoal sequence, and derive the likelihood of a subgoal sequence, given an observed state sequence. Then we describe the nonparametric model of subgoal sequences, and a Markov chain Monte Carlo (MCMC) method for jointly inferring the number and composition of the subgoal sequences underlying a series of action sequences. Finally, we show how to use subgoal sequences learned from previous observations to predict the subgoals and destination of a novel partial action sequence.

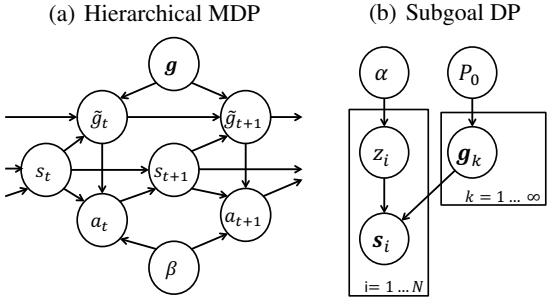

Figure 1: Graphical models for our framework. (a) Hierarchical MDP planning model of state sequences. Agents select actions according to a probabilistic policy in the hierarchical MDP defined by the subgoal sequence $\boldsymbol{g}$. β is a parameter for soft-max action selection. (b) The Bayesian nonparametric subgoal model takes the form of a Dirichlet process (DP), in which each action sequence depends on a goal sequence sampled from a DP with concentration parameter α.

Subgoal sequence likelihood

Our hierarchical MDP formulation is closely related to the options framework for hierarchical reinforcement learning (Sutton, Precup, and Singh 1999). For simplicity, we assume that actions and state transitions are deterministic, that each action incurs a cost of 2, that the discount factor $\gamma = 1.0$, and that the destination yields a reward of 100 once all subgoals have been achieved.

The subgoal sequence $\boldsymbol{g}$ is analogous to a set of options, with initiation and termination conditions that require each subgoal to be achieved in sequential order. In Fig. 1(a), the variable $\tilde{g}_t$ keeps track of the current subgoal at time t. Assume the current subgoal is $\tilde{g}_t = g_m$; if the agent reaches the current subgoal, i.e., $s_t = \tilde{g}_t$, then $\tilde{g}_{t+1} \leftarrow g_{m+1}$ should be the next subgoal, otherwise the subgoal should stay the same ($\tilde{g}_{t+1} \leftarrow \tilde{g}_t$).

Based on this, an observed sequence $\boldsymbol{s}$ can be divided into multiple segments corresponding to $\boldsymbol{g}$. We define boundary b_m to be the first timestep t after g_{m-1} is achieved: $b_m = \min(\{t | s_{t-1} = g_m \wedge t > b_{j-1}\})$; $b_0 = 1$. We write the boundary vector $\boldsymbol{b} = \langle b_0, b_1, ... \rangle$. If $\boldsymbol{s}$ achieves all subgoals in $\boldsymbol{g}$ in order, the length of $\boldsymbol{b}$, $\dim(\boldsymbol{b})$ should be $\dim(\boldsymbol{g}) + 1$. Otherwise, $\boldsymbol{s}$ does not satisfy $\boldsymbol{g}$, so $P(\boldsymbol{s}|\boldsymbol{g})$ should be 0. The subgoal sequence likelihood is then:

$$P(\boldsymbol{s}|\boldsymbol{g}) = \begin{cases} \prod_{m=1}^{\dim(\boldsymbol{g})} \prod_{t=b_{m-1}}^{b_m - 1} P(s_{t+1}|s_t, g_m); \\ \quad (\text{if } \dim(\boldsymbol{b}) = \dim(\boldsymbol{g}) + 1) \\ 0; (\text{otherwise}), \end{cases} \quad (1)$$

where $P(s_{t+1}|s_t, g_m) = \sum_{a_t \in A} P(s_{t+1}|s_t, a_t) P(a_t|s_t, g_m)$ is the marginal probability of the state transition from s_t to s_{t+1}, integrating over actions. $P(a_t|s_t, g_m) \propto \exp(\beta Q_{g_m}(s_t, a_t))$ is a softmax policy of the local MDP state-action value function for subgoal g_m, based on the assumption that the observed agent plans approximately rationally, stochastically maximizing expected reward and minimizing cost.

A similar likelihood computation was used by (Michini, Cutler, and How 2013) within an MCMC method for inferring subgoal sequences from user demonstrations. However, this approach focused on learning only one subgoal sequence from one action sequence; in the next section, we describe a nonparametric Bayesian model and MCMC methods for inferring multiple subgoal sequences, given a series of state sequences.

Nonparametric subgoal inference

We now consider inference of subgoal structure by observing multiple sequences for a certain destination. We denote the set of N behavior sequences as $\boldsymbol{s}_{1:N}$ and the ith sequence as $\boldsymbol{s}_i$. We denote a set of K subgoal sequences as $\boldsymbol{g}_{1:K}$, and the kth sequence as $\boldsymbol{g}_k$. The problem of nonparametric subgoal inference is to compute $P(\boldsymbol{g}_{1:K}|\boldsymbol{s}_{1:N})$ for an unbounded number of sequences K.

We model the set of unknown subgoal sequences using a nonparametric Bayesian model, which allows us to con-

sider an unbounded number of subgoal sequences for each destination. We use the Dirichlet process (DP) to express the distribution over subgoal sequences, following (Buchsbaum et al. 2015). A graphical model of our DP model is shown in Fig. 1(b). We use the Chinese Restaurant Process (CRP) representation to efficiently draw samples from the DP. First, the CRP selects a "table" for each observation $\boldsymbol{s}_i$, conditioned on all previous table assignments and the concentration parameter α_0. z_i is the index of the table assigned to state sequence $\boldsymbol{s}_i$. Next, for each CRP table, a subgoal sequence is sampled from the base distribution P_0, and $\boldsymbol{g}_k$ denotes the subgoal sequence associated with the kth table. The state sequence $\boldsymbol{s}_i$ is then generated given its associated subgoal sequence.

We use a MCMC method to compute the posterior probability over subgoal sequences, specifically, Gibbs sampling. Gibbs sampling allows us to approximate the complex DP distribution by inducing a Markov chain over samples from the CRP. Gibbs sampling over the CRP is a standard MCMC algorithm for DP inference (Neal 2000). Algorithm 1 is an overview of our algorithm. As an initialization step, we assign a different table for each state sequence, and draw subgoal sequences from the conditional distribution over subgoal sequences given the state sequence assigned to each table. We then repeat the table re-assignment step (resampling the table for each state sequence) and the parameter re-assignment step (drawing the subgoal sequences from the conditional distribution over subgoal sequences, given all sequences assigned to a table). For the table re-assignment step, we calculate the probability $P(z_i = k|z_{-i}, \boldsymbol{s}_i)$ to assign sequence $\boldsymbol{s}_i$ to table k according to standard Gibbs sampling for the CRP:

$$P(z_i = k|z_{-i}, \boldsymbol{s}_i) = \begin{cases} \frac{n_{-i,k}}{N-1+\alpha} P(\boldsymbol{s}_i|\boldsymbol{g}_k) \\ (\text{If } k = z_j \text{ for some } i \neq j) \\ \frac{\alpha}{N-1+\alpha} \int P(\boldsymbol{s}_i|\boldsymbol{g}) P_0(\boldsymbol{g}) d\boldsymbol{g} \\ (\text{If } k \neq z_j \text{ for all } i \neq j), \end{cases} \quad (2)$$

where z_{-i} denotes table assignments, excluding sequence i, and $n_{-i,k}$ denotes the number of sequences assigned to table k, excluding sequence i.

However, for our problem, $P(\boldsymbol{s}_i|\boldsymbol{g})$ is the MDP likelihood of a subgoal sequence. Because this is a non-conjugate distribution, we cannot integrate this equation analytically. If the environment is small, we can enumerate all of $\boldsymbol{g}$, and compute it directly as in the previous section. In large environments we must use an approximate method to choose a new table; some techniques are described by (Neal 2000). In the parameter re-assignment step, we draw the subgoal sequence from the posterior over subgoal sequences, given all sequences assigned to a table. Assume $\boldsymbol{s}_{1:l}$ are the sequences assigned to table k. The distribution to draw a new subgoal sequence $\boldsymbol{g}$ for table k should be $P(\boldsymbol{g}|\boldsymbol{s}_{1:l})$. This probability for each subgoal sequence can be calculated as follows:

$$P(\boldsymbol{g}|\boldsymbol{s}_{1:l}) \propto P_0(\boldsymbol{g}) P(\boldsymbol{s}_{1:l}|\boldsymbol{g}) = P_0(\boldsymbol{g}) \prod_{i=1}^{l} P(\boldsymbol{s}_i|\boldsymbol{g}) \quad (3)$$

At the end of each step of the loop, we count the number of subgoal sequences for each state sequence. We represent the number of times that $\boldsymbol{g}$ is assigned any state sequence as $\mathbf{c}(\boldsymbol{g})$. The normalized count corresponds to $P(\boldsymbol{g} \in \boldsymbol{g}_{1:K}|\boldsymbol{s}_{1:N})$.

Algorithm 1 Subgoal inference

for $i = 1$ to N **do**
  $z_i = i; \boldsymbol{g}_i \sim P(\boldsymbol{g}_i|\boldsymbol{s}_i)$ // Initialize Step (See, Eq. 1)
end for
for $r = 1$ to $repeat$ **do**
  for $i = 1$ to N **do**
    $z_i \sim P(z_i|z_{-i}, \boldsymbol{s}_i)$ // Table Re-assign Step (See, Eq. 2)
    if z_i is index for new table **then**
      $\boldsymbol{g}_{z_i} \sim P(\boldsymbol{g}_{z_i}|\boldsymbol{s}_i)$ // Parameter Initialize for new table (See, Eq. 1)
    end if
  end for
  for $k \in \{z_1, z_2, ...\}$ **do**
    $\boldsymbol{g}_k \sim P(\boldsymbol{g}_k|\{\boldsymbol{s}_i|1 \leq i \leq N, z_i = k\})$ // Parameter Re-assign Step (See, 3)
  end for
  for $k \in \{z_1, z_2, ...\}$ **do**
    $\mathbf{c}(\boldsymbol{g}_k) \leftarrow \mathbf{c}(\boldsymbol{g}_k) + 1$
  end for
end for
for all $\{\boldsymbol{g}|\mathbf{c}(\boldsymbol{g}) > 0\}$ **do**
  $P(\boldsymbol{g} \in \boldsymbol{g}_{1:K}|\boldsymbol{s}_{1:N}) \leftarrow \mathbf{c}(\boldsymbol{g})/(repeat)$
end for
output $P(\boldsymbol{g} \in \boldsymbol{g}_{1:K}|\boldsymbol{s}_{1:N})$

Experiments

Our two experiments presented a "warehouse" scenario involving the delivery of various items to destinations in the environment shown in Fig. 2. The warehouse has three delivery destinations: A, B, and C. There are nine potential items to be delivered, marked by numbers 1-9. The items are arranged into three rows, and for each delivery, one item can be delivered from each row.

Each job in the warehouse has a specific destination, and several possible "item lists" to deliver. An item list consists of either one, two, three items that must be delivered to the destination. For each delivery, one of these item lists is selected by the warehouse scheduler. In addition to items from the current item list, workers are encouraged to pick up "Add-on" items , but only if this won't increase the number of steps on their path to the destination. For example, if the item list for Fig. 2(b) is [2,8], item 5 is a good Add-on item for that delivery, because item 5 is the only additional item that does not require additional steps to obtain, given the start point, item list, and destination. There is a direct correspondence between delivery destinations and item lists in this setting and destinations and subgoals as represented by our model.

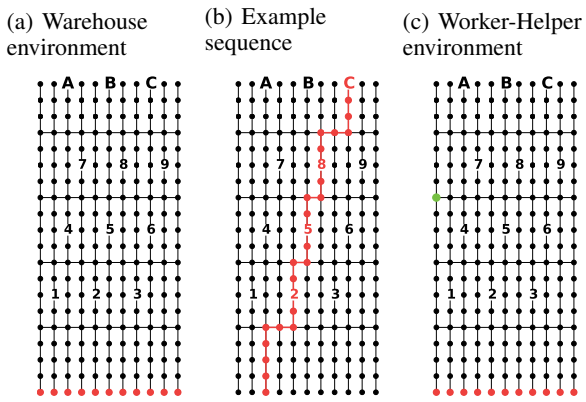

Figure 2: (a) Experimental scenario. A, B, and C mark possible destinations, and numbers 1-9 mark items that workers must deliver. Red points mark potential starting locations.(b) Example agent behavior. (c) Environment of Experiment 2. Green point is the starting location of the Helper.

Experiment 1: Subgoal inference

Participants We recruited participants for this study using Amazon Mechanical Turk. Participants were 29 adults located in the USA (Male 13, Female 14, Unknown 2). Mean age was 32 years old.

Procedure First, in a training phase, subjects learned to perform an example warehouse job to learn the rules of the scenario. In each trial of the testing phase, participants saw 8 paths for a particular job, and inferred how many item lists the job had, and which items were on each list.

Stimuli We prepared 22 jobs for the testing phase, including a range of subgoal structures of varying complexity and difficulty. First were jobs with one item list, involving 1, 2, or 3 subgoals. Second were jobs with two item lists, involving 1, 2, or 3 subgoals for both item lists, or 1 and 3 subgoals for each item list, respectively. Some jobs presented two types of action sequences. In one type, the action sequence could reach all subgoals without deviating from the shortest path to the destination. These cases were difficult in that they did not provide direct evidence for any subgoal. In the other type, the action had to make a detour to reach the subgoal, which provided stronger evidence for that subgoal.

Modeling In our MDP representation of the environment, $\mathcal{S}$ corresponds to the agent location, and $\mathcal{A}$ is the set of movements (up, left, and right). We used a uniform distribution for the prior over subgoal sequences $P_0(\boldsymbol{g})$ and set $\beta = 6$. For the nonparametric Bayesian model, we set the concentration parameter $\alpha_0 = 0.015$. We used 5,000 iterations of Gibbs sampling, with 1,000 steps burn-in time to reach the stationary distribution.

Alternative models We prepared 3 alternative models to compare with our Bayesian nonparametric model: *Independent model*, *Logical possibility model*, and *Copy model*. To explain each alternative model, we consider the case in which the model observes N sequences $\boldsymbol{s}_{1:N}$.

The Independent model is most similar to our approach. The only difference is that this model does not use the CRP, but instead calculates the posterior for each sequence $P(\boldsymbol{g}_i|\boldsymbol{s}_i)$ independently. The joint probability for all sequences is given by: $P(\boldsymbol{g}_{1:N}|\boldsymbol{s}_{1:N}) = \prod_{i=1}^{N} P(\boldsymbol{g}_i|\boldsymbol{s}_i)$. The marginal probability for each subgoal sequence is given by: $P(\boldsymbol{g} \in \boldsymbol{g}_{1:N}|\boldsymbol{s}_{1:N}) = \sum_{\{\boldsymbol{g}_{1:N}|\boldsymbol{g}\in\boldsymbol{g}_{1:N}\}} P(\boldsymbol{g}_{1:N}|\boldsymbol{s}_{1:N})$.

The Logical possibility model is a more heuristic approach, which computes the proportion of observations in which a subgoal sequence is present within the action sequence. Formally, $P(\boldsymbol{g} \in \boldsymbol{g}_{1:K}|\boldsymbol{s}_{1:N}) = |\{i|1 \leq i \leq N \wedge \boldsymbol{g} \subset \boldsymbol{s}_i\}|/N$.

The Copy model is the simplest alternative, but perhaps the most intuitive. It assigns probability 1 to the subgoals which have the highest likelihood for each observed sequence. In other words, the inferred subgoal sequence is the maximal subgoal sequence that is included by an observed action sequence. If we represent such a subgoal sequence $MaxSubgoal(\boldsymbol{s})$, the copy model can be written: $P(\boldsymbol{g} \in \boldsymbol{g}_{1:K}|\boldsymbol{s}_{1:N}) = 1$ if $\boldsymbol{g} = MaxSubgoal(\boldsymbol{s}_i)$ for some i, otherwise 0.

Results To compare human behavioral results with those of our models, we compute the proportion of participants who selected each item list for each trial. Model predictions were based on the marginal probability that each item list generated at least one path.

Table 1 compares the correlation of the Bayesian nonparametric subgoal model and our three alternative models with human inferences. The Bayesian nonparametric model correlates very strongly with human inferences. The other models also correlate positively with human inferences, but all are substantially worse than the Bayesian nonparametric model.

Model	CRP	Independent	LP	Copy
Correlation	0.973	0.644	0.267	0.495

Table 1: Pearson correlation between human inferences and computational models. LP stands for "Logical Possibility".

Fig. 3 plots human inferences against the Bayesian nonparametric model predictions for all trials and subgoal sequences. This shows that our model can almost perfectly predict whether humans will infer that a certain item list is included in each job. For a small number of trials, humans are somewhat uncertain whether to include a particular item list, and the model captures this uncertainty as well.

Finally, we compare our model predictions with human inferences for two specific jobs. In Fig. 4(a), each path includes item 5, but there are no paths which include only this item. Humans and the Bayesian nonparametric model infer that [5] is the most likely subgoal sequence, based on the systematic deviation from the shortest path to the destination to reach this item. All other models fail in revealing ways, inferring more item lists than are necessary to explain the data. In Fig. 4(b), there are two item lists. Although ev-

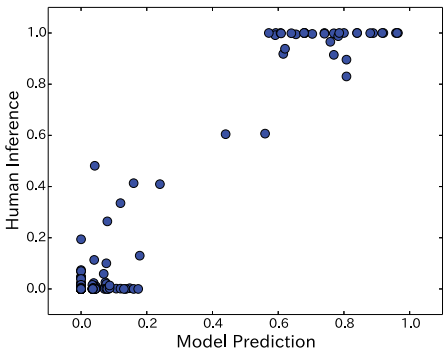

Figure 3: Scatter plot of human inferences versus Bayesian subgoal model predictions (r = 0.973).

ery path is an efficient route to the destination, without inferring an item list, humans and our model can infer these two subgoal sequences correctly by assessing the probability that these items are selected intentionally rather than by coincidence. Our alternative models once again fail to capture people's judgments.

Experiment 2: User support using subgoal inference

Task definition We call this the **Worker-Helper Task**. The main environment is the same as in Experiment 1, but there are two types of agents. One type is **Worker**, whose actions are structured identically to those presented in Experiment 1. The Worker's job remains the same: to deliver an item list to a destination. The other type is **Helper**, which must support Workers in achieving their goals. The task of the Helper is to learn the structure of the Worker's jobs, then use this to help the Worker complete a job in progress by retrieving an item from the item list of the Worker, and delivering it to the destination. Helpers begin each trial on the left side of the warehouse, midway between the second row of items (4, 5 and 6) and the third row of items (7, 8 and 9). Fig. 2(c) illustrates this task, with the start point of the Helper marked by a green dot.

Collaboration protocol On each trial, the Worker randomly chooses a destination (A, B, or C) and an item list associated with that destination. The set of item lists for each destination is fixed and known to the Worker, but not the Helper. Next, the Worker plans a path to achieve their goals and begins to move through the warehouse. The Helper observes the Worker's behavior and decides which target item to retrieve by inferring the Worker's subgoal sequence (due to the Helper's starting location, the target item will always be in the third row, i.e., item 7, 8, or 9). Once the Helper decides their target item, the Helper shows this target to the Worker and begins to move. After the Worker observes the Helper's target item, the Worker re-plans its path under the assumption that the Helper will get the target item. After the Helper gets the target item, the Helper moves toward the destination inferred by observing the Worker's path. When

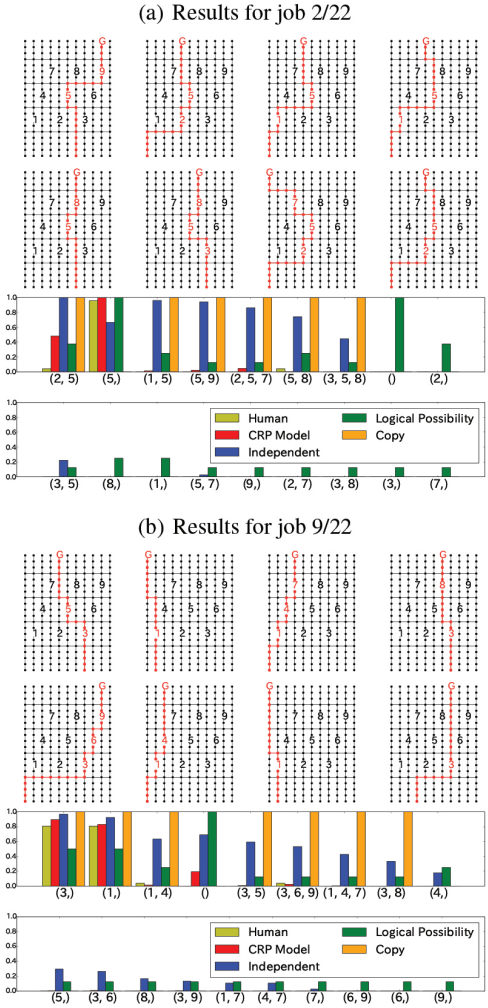

Figure 4: Example results for two jobs. (a) Job with item list [5]; humans and CRP model infer [5] is the most probable subgoal sequence. Alternative models fail to predict human judgments. (b) Job with two item lists: [1] and [3]. Humans and CRP model correctly infer these two subgoal sequences. Once again, the alternative models fail to predict human judgments.

the Helper cannot decide on a target item, they do nothing.

Modeling We assume that workers take the optimal actions, given their subgoal sequence, i.e., $a_t = \arg\max_{a_t} Q_{\boldsymbol{g}}^{\pi^*}(s_t, a_t)$. The Helper estimates the marginal probability that each item is included in the subgoal sequence which the Worker is currently following. The Helper computes this probability based on the partial path of the Worker $s_{1:t}$, and n previously observed paths $\boldsymbol{s}_{1:n}$, under the assumption that the Worker plans approximately rationally, with softmax parameter $\beta = 2$. When the probability of a target exceeds a certain value, the Helper decides on this item. The Worker then re-plans by removing the target item and the Helper begins to take optimal actions, as does the

Worker.

Probability of target item and destination The marginal probability of target item g', given $s_{1:t}$ and $\boldsymbol{s}_{1:n}$ is: $P(g'|s_{1:t}, \boldsymbol{s}_{1:n}) = \sum_d \sum_{\{\boldsymbol{g}|g' \subseteq \boldsymbol{g}\}} P(s_{1:t}|\boldsymbol{g}, d) P(\boldsymbol{g} \in \boldsymbol{g}_{1:K}|\boldsymbol{s}^d_{1:n})$, where $\boldsymbol{s}^d_{1:n}$ is the subset of previously observed paths with destination d. $P(\boldsymbol{g} \in \boldsymbol{g}_{1:K}|\boldsymbol{s}^d_{1:n})$ is calculated using the subgoal inference method in Section 2. $P(s_{1:t}|\boldsymbol{g}, d)$ can be computed using Eq. 1, but without the constraint that $\dim(\boldsymbol{g}) = \dim(\boldsymbol{g}) + 1$. The marginal probability of destination d, given $s_{1:t}$ and $\boldsymbol{s}_{1:n}$ is: $P(d|s_{1:t}, \boldsymbol{s}_{1:n}) \propto \sum_{\boldsymbol{g}} P(s_{1:t}|\boldsymbol{g}, d) P(\boldsymbol{g} \in \boldsymbol{g}_{1:K}|\boldsymbol{s}^d_{1:n})$.

Alternative models We use the same alternative models as in Experiment 1, and a common collaboration protocol (described above) for each subgoal inference model. We also add two new models. One is No Helper, which means the Helper has no subgoal knowledge. As a result, the Helper does nothing. The other is the Ground Truth model, which means the Helper knows the correct subgoal. It is a benchmark to measure the benefit of the Bayesian model.

Test method We tested three natural types of subgoal settings. In the first, each of the three destinations has one subgoal sequence, which consists of either 7, 8, or 9. In the second, each of the three destinations has one subgoal sequence, which consists of either item 4, 5, or 6 and either item 7, 8, or 9. In the third, each of the three destinations has two subgoal sequences which consist of either item 7, 8, or 9. We generate every possible combination of items for each subgoal setting, then we learn many subgoal structures and evaluate the performance for each subgoal structure.

Evaluation for one subgoal structure We generate a number of sequences of Workers from random start points for each destination. We then compute subgoal inferences for each model using the sequences. We then execute the collaborative task 99 times for each subgoal inference (11 points times 9 trials), and evaluate the performance of Workers in achieving their destination. We repeat these tests 5 times for each subgoal structure. We use the following score to measure performance. If the Worker achieves their goal, they score 100 points, but each action costs 2 points. This directly corresponds to the MDP reward and cost settings in Experiment 1.

Results We executed the test, varying the number of input sequences to evaluate the dependence of each model on the amount of input data. Fig. 5(a) shows the average score of each model as a function of the number of input sequences. Generally, the Helper using the nonparametric Bayesian model is more beneficial than any alternative model. This model achieves good performance (almost as high as the Ground Truth model) using even a small number of input sequences.

Further, because the alternative models depend highly on the input sequences, the scores of these models are unstable. Fig. 5(b) shows the average variance of the results for each

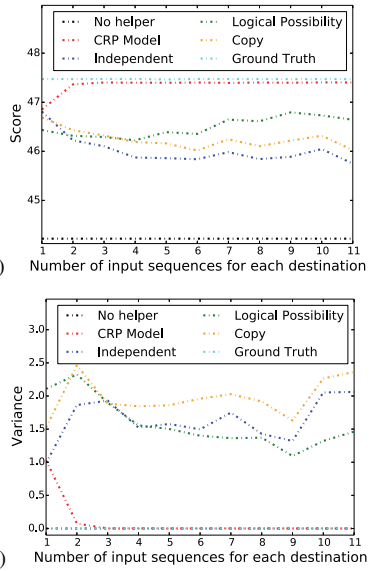

Figure 5: Model scores (a) and variance (b) as a function of the number of input sequences. Variance of No helper and Ground Truth is always 0.

subgoal setting (we repeated the experiment 5 times for each subgoal setting; this score is the variance of these). None of the alternative models can provide stable user support. In contrast, the nonparametric Bayesian model provides stable support using even a small number of input sequences (over two input sequences). This stability is a key factor for user support, since unstable help can confuse and frustrate users.

Conclusion

We presented a model of how humans infer subgoals from observations of complex action sequences. This model used rational hierarchical planning over subgoal structures generated by a nonparametric Bayesian model to capture people's intuitions about the structure of intentional actions. We showed how Bayesian inference over this generative model using a novel MCMC method predicts quantitative human subgoal inferences with high accuracy and precision. We then showed that our model is useful in practice, enhancing performance in an application to an artificial user support task.

Our modeling and experimental scenarios are extremely simplistic in comparison with real human behavior. One limitation is our assumption that subgoal sequences are chosen according to probabilities determined by the Dirichlet Process. More generally, the probability of choosing a particular subgoal sequence will depend on the efficiency of that subgoal sequence, relative to the alternatives. To enhance the expressiveness of our model, Infinite PCFGs (Liang et al. 2007), Adaptor Grammars (Johnson, Griffiths, and Goldwater 2007), or fragment grammars (ODonnell 2015) are promising extensions to the simple Dirichlet Process we employ. These and other frameworks which can be applied to structured goal representation complement our work here by

naturally interfacing with models of hierarchical planning analogously to the model we describe.

Acknowledgments

This work was supported by the Center for Brains, Minds & Machines (CBMM), funded by NSF STC award CCF-1231216, and by NSF grant IIS-1227495.

References

Baker, C. L.; Saxe, R.; and Tenenbaum, J. B. 2009. Action understanding as inverse planning. *Cognition* 113:329–49.

Baldwin, D.; Andersson, A.; Saffran, J.; and Meyer, M. 2008. Segmenting dynamic human action via statistical structure. *Cognition* 106(3):1382–1407.

Buchsbaum, D.; Griffiths, T. L.; Plunkett, D.; Gopnik, A.; and Baldwin, D. 2015. Inferring action structure and causal relationships in continuous sequences of human action. *Cognitive Psychology* 76:30–77.

Bui, H. H.; Venkatesh, S.; and West, G. 2002. Policy Recognition in the Abstract Hidden Markov Model. *Journal of Artificial Intelligence Research* 17:451–499.

Charniak, E., and Goldman, R. 1993. A Bayesian model of plan recognition. *Artificial Intelligence* 64(1):53–79.

Dennett, D. C. 1987. *The Intentional Stance.* MIT Press, Cambridge, MA.

Gergely, G.; Nádasdy, Z.; Csibra, G.; and Bíró, S. 1995. Taking the intentional stance at 12 months of age. *Cognition* 56(2):165–193.

Johnson, M.; Griffiths, T. L.; and Goldwater, S. 2007. Adaptor grammars: A framework for specifying compositional nonparametric Bayesian models. In *Advances in Neural Information Processing Systems*, 641–648.

Kautz, H. A., and Allen, J. F. 1986. Generalized Plan Recognition. In *Proceedings of the 5th National Conference on Artificial Intelligence*, 32–37.

Liang, P.; Petrov, S.; Jordan, M. I.; and Klein, D. 2007. The Infnite PCFG using Hierarchical Dirichlet Processes. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, 688–697.

Michini, B.; Cutler, M.; and How, J. P. 2013. Scalable reward learning from demonstration. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 303–308.

Neal, R. M. 2000. Markov Chain Sampling Methods for Dirichlet Process Mixture Models. *Journal of Computational and Graphical Statistics* 9(2):249–265.

ODonnell, T. J. 2015. *Productivity and Reuse in Language: A Theory of Linguistic Computation and Storage.* MIT Press, Cambridge, MA.

Ramírez, M., and Geffner, H. 2010. Probabilistic Plan Recognition Using Off-the-Shelf Classical Planners. In *Proceedings of the 24th AAAI Conference on Artificial Intelligence*, 1121–1126.

Schachner, A., and Carey, S. 2013. Reasoning about irrational actions : When intentional movements cannot be explained , the movements themselves are seen as the goal. *Cognition* 129(2):309–327.

Schank, R. C., and Abelson, R. P. 1977. *Scripts, Plans, Goals, and Understanding: An Inquiry Into Human Knowledge Structures.* Lawrence Erlbaum Associates, Hillsdale, NJ.

Sutton, R. S.; Precup, D.; and Singh, S. 1999. Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning. *Artificial Intelligence* 112(1-2):181–211.