Using Bayesian Inference to Learn High-Level Tasks from a Human Teacher

***Abstract*— Humans learn from teachers by observing gestures, reinforcements, and words, which we collectively call *signals*. Often a new signal will support a different interpretation of an earlier signal, resulting in a different belief about the task being learned. If robots are to learn from these signals, they must perform similar inferences. We propose the use of Bayesian inference to allow robots to learn tasks from human teachers. We review Bayesian inference, describe its application to the problem of learning high-level tasks from a human teacher, and work through a specific implementation on a robot. Bayesian inference is shown to quickly converge to the correct task specification.**

I. INTRODUCTION

The ability of robots to perform meaningful work in complex, real-world environments is continually expanding [1], [2], [3]. To take advantage of these abilities, robot users will need a mechanism for defining their own tasks. Many of these users will not be proficient programmers, but will be familiar with teaching other humans through gestures, reinforcements, and words, which we collectively call "signals". It would be useful if they could teach robots using these same signals. The problem is that a signal can often have multiple interpretations, either because of perception errors or because the gesture, reinforcement, or word by itself does not carry enough information to fully specify the task. These multiple interpretations result in multiple task specifications. Fortunately, as more signals arrive, fewer interpretations "make sense" and the task becomes clearer. We propose the use of Bayesian inference to give robots this type of reasoning, allowing users to specify new tasks using familiar signals.

We assume that the robot has been pre-programmed with a set of primitive actions. The goal is to learn a composition of these primitive actions, which we call a "task" or "high-level task" to emphasize the use of primitive actions. We are concerned with the *teaching* of the task, not the commanding of the task, which may also make use of Bayesian inference.

In this paper we review Bayesian inference, describe its application to task learning, and work through an example of an actual robot learning a simple task. But first we start by reviewing some related work.

II. RELATED WORK

Multiple researchers have addressed the topic of learning from a human teacher and many make use of Bayesian inference. We decompose these works into two groups based on the type of problem addressed: control vs. communication.

In a "control problem", the robot needs a mapping from sensor values to actuator controls. This mapping can be difficult to specify. Often a human can directly control the robot to perform the behavior, even though they are unable to write down the function they are using. To make use of the human's ability, a log is captured of the sensor values and control inputs as the human controls the robot. This log is then used to "learn" the mapping. This is often called imitation learning or apprenticeship learning. Some examples of control problems that have been addressed in this way are: performing helicopter aerobatics [4], navigating a corridor [5], [6], [7], and pushing obstacles [8]. This type of learning could be used to create the primitive actions that are assumed in this paper.

In what we are calling a "communication problem", the human can write down exactly what they want the robot to do. More specifically, the human can write down the sequence of primitive actions that make up a task. The problem is communicating this sequence to the robot. While a programmer could easily "code" this in, for a non-technical user, this is a communication problem where the robot must extract the specification of a task from human signals.

Most of the work addressing this communication problem treats the teacher signals deterministically. For example, when the human gives a signal, the primitive action currently executing gets appended to the primitive action sequence making up the task [9], [10], [11], [12]. But, as we mentioned before, signals can often be interpreted in multiple ways, and it is only after the information from several signals is integrated that the meaning of early signals becomes clear. Because of the continually accumulating signal information, inference is needed.

To our knowledge, no one has applied Bayesian inference to this problem. The following works maintain counts for primitive actions and then use these counts to form the task, but no formal inference is performed [13], [14], [15].

In [16], reinforcement learning, specifically Q-Learning [17], is applied to this problem. The authors identify the tendency of humans to "shape" when they are teaching tasks, and encourage future work to incorporate human shaping into reinforcement learning algorithms. Their suggestion is to add another button for the teacher to indicate when they are "shaping", thereby removing the uncertainty. This solution will work for the reinforcement learning case, where a reinforcement is the only signal, but does not scale well to multiple signal types. This type of modification to the teacher signal is unnecessary with Bayesian inference, as we show in our demonstration below.

A technique which bears similarity to our research is Bayesian robot programming (BRP) [8], in which the authors

use probability to address traditional programming. Instead of defining the preconditions for switching from one behavior to another, under BRP the programmer specifies what the robot is likely to see when it is executing a behavior. Bayes rule is then used to express the distribution over which behavior should be run in terms of what is likely to be seen for each behavior. This is not the use of Bayesian inference we propose in this paper, i.e. inferring task specifics from a human teacher.

Also, in our demonstration below, we use a teacher applied reinforcement as a signal. Many works have used teacher applied reinforcements [16], [14], [15], [10], [11], etc., but none with Bayesian inference.

III. BAYESIAN INFERENCE

Bayesian inference is a technique for estimating unobservable quantities from observable quantities. In this section we give an overview of Bayesian inference, beginning with some definitions.

We use $p(\cdot)$ as notation for both probability density functions and probability mass functions. $p(X)$ is shorthand for $p(X = x)$ where x is some event in the domain of the random variable X. If X is a time varying random variable, we use X_t to indicate the value of X at time t and we define the shorthand X^t to mean $(X_t, X_{t-1}, \cdots, X_1)$. Finally, we define $\hat{X}^t$ to be X^t with X_0 added on.

In its simplest form, Bayesian inference is just Bayes rule. Bayes rule allows you to update your belief about a hidden quantity H given an observed quantity O, and is defined as,

$$p(H|O) = p(O|H) \times p(H)/p(O) \quad (1)$$
$$\propto p(O|H) \times p(H) \quad (2)$$

$p(O|H)$ specifies the probability of measuring o given $H = h$. It is called the "measurement model" and is generally easier to specify than $p(H|O)$, which is why Bayes rule is useful. $p(H)$ is called the prior distribution over H and represents the belief about H before O is measured. The second line follows from the first and the fact that $p(O)$ is a constant since we know the value of O. $p(H|O)$ is called the posterior distribution.

In the case where H and O vary with time, the posterior distribution $p(\hat{H}^t|O^t)$ can be decomposed as follows,

$$p(\hat{H}^t|O^t) \propto p(O_t|\hat{H}^t, O^{t-1}) \times p(\hat{H}^t|O^{t-1}) \quad (3)$$
$$= p(O_t|\hat{H}^t, O^{t-1}) \times p(H_t|\hat{H}^{t-1}, O^{t-1}) \times p(\hat{H}^{t-1}|O^{t-1}). \quad (4)$$

Eq. (3) is a direct application of Bayes rule. Eq. (4) follows from Eq. (3) and the definition of conditional probability. $p(O_t|\hat{H}^t, O^{t-1})$ is again the measurement model. $p(H_t|\hat{H}^{t-1}, O^{t-1})$ is called the "motion model" and specifies the motion of the time varying, hidden random variable H. In the context of an update to the posterior distribution, $p(\hat{H}^{t-1}|O^{t-1})$, which is the posterior distribution from the previous time step, is also called the prior distribution. We try to make it clear when we mean the prior distribution from the previous time step, or the prior distribution at time zero, $p(H_0)$.

Given a set of measurements O^t and assuming H is discrete, one way to compute $p(\hat{H}^t|O^t)$ for a specific assignment to the $(H_t, H_{t-1}, \cdots, H_1, H_0)$ is to start from $p(H_0)$ and recursively apply Eq. (4). This process can be done for each of the N^{t+1} assignments to the $(H_t, H_{t-1}, \cdots, H_1, H_0)$, where N is the number of values that H can take on. The complete set of N^{t+1} assignments is called the posterior space. We will discuss techniques for dealing with the exponential growth of the posterior space in the "Discussion" section bellow.

In general, Bayesian inference can be applied to problems with any number of discrete or continuous, time varying or static, hidden and observed, random variables, e.g. $p(H^t, I, J, \ldots|O^t, P, Q^t, \ldots)$. As before, the application of Bayes rule followed by the definition of conditional probability can be used to decompose this posterior. In general, in order to use Bayesian inference the following must be specified:

Necessary Components for Bayesian Inference

1) Define the **hidden random variables**.
2) Define the **observable random variables**.
3) Specify a **measurement model** for each observable random variable.
4) Specify a **motion model** for each time varying hidden random variable.
5) Specify the **prior distribution** over the hidden random variables at time zero.

Next we frame high-level task learning as a Bayesian inference problem.

IV. BAYESIAN INFERENCE APPLIED TO LEARNING HIGH-LEVEL TASKS

The problem of learning high-level tasks from a human teacher involves inferring task details from signals emitted by the teacher. In order to apply Bayesian inference we need to specify the hidden random variables, the observed random variables, the measurement models, any motion models, and the prior distribution.

For the problem of learning high-level tasks from human teachers, the hidden random variables capture what needs to be specified about the task, a complete assignment to the hidden random variables should allow the robot to perform the task. The observed random variables capture useful information with regards to the signals from the teacher, such as the direction the teacher was pointing or the location of objects in the room.

Measurement models need to be specified that predict these measurement values given a complete assignment to the hidden random variables. For some of these measurements the link between the measured value of the observed random variable and the hidden random variables specifying the task could be complicated, making the specification of the

measurement model a difficult task. In many situations this link is simplified if additional hidden random variables are introduced. Thus, introducing additional hidden random variables can be useful even if those variables do not directly specify parameters of the task. An example of this technique is provided in our demonstration below. Some of the hidden random variables will vary with time and for these a motion model should be specified. Finally a prior distribution over the hidden random variables defining the task should be specified.

Once these have been specified, Bayesian inference techniques can be used to incorporate new signals and update the posterior distribution over the hidden random variables defining the task.

V. ACTION SELECTION

This paper focuses on the *inference* of task details from human signals, but often the robot will need to move around in order to elicit signals from the human teacher. This sections gives suggestions as to how the inference described in this paper can help with *autonomously* selecting actions during the teaching process.

With each new teaching signal, Bayesian inference updates the distribution over the hidden random variables defining the task, but from the start this distribution exists. One option for action selection during teaching is to sample from the current distribution and then perform the task according to the sample. As the distribution converges to the desired task, the robot will more often choose samples that execute the task correctly. Also, by sampling, instead of choosing the mean or mode, randomness is introduced and the robot naturally explores the task space in proportion to its current belief. A variation on this is used in the demonstration below.

Direct control of the robot during the teaching process is another option, but we feel that when using Bayesian inference, autonomous action selection during teaching will often be necessary, since it will not always be evident to the teacher which details of the task remain unclear to the robot.[1]

VI. DEMONSTRATION

We now detail the use of Bayesian inference to learning a specific task. This demonstrations is meant to give the reader the flavor of how to apply Bayesian inference. The random variables, measurement models, and motion models described are based on intuition, we make no claims about their accuracy. It is likely that the effectiveness of this technique would be improved with more accurate models. However, the focus of this paper is to establish the basis of Bayesian inference.

[1]Some of this uncertainty might be communicated using emotional projecting techniques from social robotics research [18], but the robot will likely still need to move itself or the world into regions where uncertainty exists.

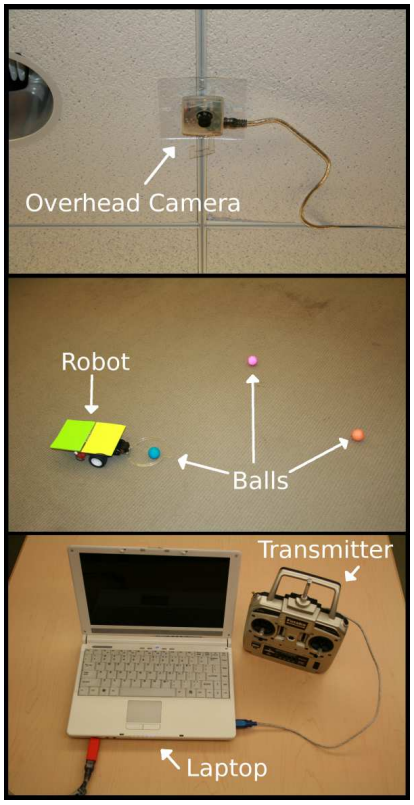

Fig. 1. This figure shows the platform used to demonstrate our approach. The top image shows the camera which tracks the robot and the three balls. The middle image shows the robot on the floor moving a ball around with its acrylic ring. The bottom image shows the laptop, which performs all of the processing, connected to a hobby transmitter. A custom circuit board was designed to interface the laptop with the transmitter. The teacher presses the spacebar on the laptop to administer a reinforcement signal.

A. Platform

The platform used in this demonstration is shown in Fig. 1. The robot measures five inches in diameter and is capable of moving three balls around a room. An overhead camera is used to track the robot and the three balls. All processing is performed on a laptop and autonomous control signals are transmitted out from the laptop over a standard remote control radio link (via a custom printed circuit board).

B. Teacher Signals

For this demonstration we use a single teacher signal. The signal is administered by pressing the spacebar on the keyboard. It is meant to indicate that the teacher approves of something the robot has done. We call this a “reinforcement signal”.

C. Task

The robot must infer from the reinforcement signals that the task consists of moving ball 1 to a distance of one foot from ball 2. This is depicted in Fig. 2. Many real-world tasks, such as retrieving a bottle of medicine or returning a toy to its basket, require a similar level of inference.

For this simple task it would be possible to create an *unambiguous* vocabulary or graphical interface so that non-

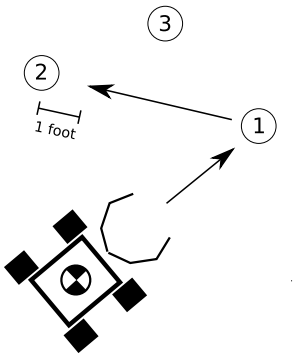

Fig. 2. The robot is taught the task of moving ball 1 to a distance of one foot from ball 2 using only reinforcement signals. Throughout the teaching session the robot acts autonomously, making use of the evidence it has acquired so far.

technical users could specify the task, but as task complexity increases, an unambiguous interface would become increasingly combersome. With Bayesian inference, a user can continue to use their familiar, *ambiguous* signals, with the increased task complexity handled through more expressive hidden random variables, motion models, and measurement models.

D. Hidden and Observable Random Variables

In this situation, a "task" for the robot consists of moving one ball to a specific distance with respect to another ball. The problem then, is to infer which ball to move, which ball to move it with respect to, and the desired separation between the balls.

The following tables define the hidden and observed random variables for this task. i is the the index of the ith reinforcement signal.

Hidden Random Variables

Mov	The ID, 0-3, of the ball that the robot should move in this task, "0" means that the robot itself should be moved.
WRT	The ID, 1-3, of the ball that ball "Mov" should be moved with respect to.
T_i	The intended type, 1-3, of the ith reinforcement signal corresponding to "final", "attention", or "distance".

Observed Random Variables

$\vec{\Theta}_i$	A vector of angles to each of the three balls when the ith reinforcement signal was received. $\vec{\Theta}_i(j)$ is the angle the robot must rotate to face ball j.
$\vec{D}_i$	A vector of the distance from the robot to each of the three balls when the ith reinforcement signal was received. $\vec{D}_i(j)$ is the distance to ball j.
H_i	The ID, 0-3, of the ball that the robot was holding when the ith reinforcement signal was received. "0" means that the robot was not holding a ball.

The T_i were introduced to simplify the definition of the measurement models for $\vec{\Theta}_i$ and $\vec{D}_i$. This will be discussed further when we talk about the measurement models. Also, as we will show later, the T_i in conjunction with the observed random variables fully define the desired distance that ball Mov should be moved to from ball WRT. This is why we did not define a hidden random variable for this distance.

The T_i attempt to capture the "type" of reinforcement signal the teacher gave: "final", "attention", or "distance". A type of "final" means that the teacher was indicating that the robot had reached the final distance. "attention" means that the teacher was pleased with the direction that the robot was facing and "distance" means that the teacher was pleased with the distance between the robot and a particular ball.

The types "attention" and "distance" are necessary because recent research has shown that human teachers do not wait for the task to be performed to perfection before administering a reinforcement signal, instead they reinforce as the robot makes small improvements towards the desired task [16]. This is called shaping, as discussed in the "related work" section.

The inference problem would be simplified if we had a button for each of the three reinforcement signal types, then the T_i would be observed random variables, and not hidden random variables which need to be inferred. We chose to use a single button for the reinforcement signal because we feel that humans may not always know why they are reinforcing the robot and because we envision future reinforcement types where no button makes sense. For example, the teacher could mistakenly press the button, in which case a reinforcement type of "mistake" would be useful.

E. Posterior Distribution & Posterior Expansion

The posterior distribution that Bayesian inference will maintain is: $p(Mov, WRT, T^i | \vec{\Theta}^i, \vec{D}^i, H^i)$.

As before, we use Bayes rule and the definition of conditional probability to decompose this posterior distribution into measurement models, motions models and the prior distribution before reinforcement signal i arrived. Here is the result of that decomposition.

$$
\begin{aligned}
p(Mov, WRT, T^i | \vec{\Theta}^i, \vec{D}^i, H^i) \propto \\
\prod_{j=1:3} p(\vec{\Theta}_i(j) | Mov, WRT, T^i, \vec{\Theta}^{i-1}, \vec{D}^i, H^i) \times \\
\prod_{j=1:3} p(\vec{D}_i(j) | Mov, WRT, T^i, \vec{\Theta}^{i-1}, \vec{D}^{i-1}, H^i) \times \\
p(H_i | Mov, WRT, T^i, \vec{\Theta}^{i-1}, \vec{D}^{i-1}, H^{i-1}) \times \\
p(T_i | Mov, WRT, T^{i-1}, \vec{\Theta}^{i-1}, \vec{D}^{i-1}, H^{i-1}) \times \\
p(Mov, WRT, T^{i-1} | \vec{\Theta}^{i-1}, \vec{D}^{i-1}, H^{i-1})
\end{aligned} \tag{5}
$$

There was flexibility as to the order in which we brought out the observed variables $\vec{\Theta}_i$, $\vec{D}_i$ and H_i. These are subjective decisions and in this case it was more useful to have $\vec{\Theta}_i$ and $\vec{D}_i$ condition on H_i than the other way around. $\vec{\Theta}_i$ and $\vec{D}_i$ are conditionally independent given H_i and the other variables, so their order did not matter.

This expansion and the specification of the models themselves, in addition to the selection of random variables is not unique; many other alternatives could have been chosen while maintaining the effectiveness of Bayesian inference.

F. Measurement Models

In the posterior decomposition above, there are three measurement models, one for each of the observed random variables: $\vec{\Theta}_i$, $\vec{D}_i$, and H_i. In this section we treat each measurement model in turn, giving rough intuition and then specifying the model.

1) $p(\vec{\Theta}_i(j)|Mov, WRT, T^i, \vec{\Theta}^{i-1}, \vec{D}^i, H^i)$*:* If the type of the new reinforcement signal is "attention", then we expect the robot to be looking in either the direction of ball Mov or in the direction of ball WRT; Mov if the robot has not yet picked up ball Mov, and WRT if it is holding ball Mov.

In all other cases the robot could be looking in any direction.

if $(T_i = attention) \cap ((Mov = j \cap H_i \neq j) \cup (H_i = Mov \cap WRT = j))$
  $p(\vec{\Theta}_i(j)|\ldots)$ is Gaussian with $\mu = 0$ and $\sigma = 20°$
else
  $p(\vec{\Theta}_i(j)|\ldots)$ is Uniform(-π,π)

Note: $\vec{\Theta}_i(j)$ takes on values in $(-\pi, \pi)$, while the domain of a Gaussian random variable is all real numbers. To account for this in the above equations we normalize the distribution by the area above pi and bellow $-pi$.

2) $p(\vec{D}_i(j)|Mov, WRT, T^i, \vec{\Theta}^{i-1}, \vec{D}^{i-1}, H^i)$*:* If the type of the new reinforcement signal is "final", then the robot must be holding ball Mov and we can predict the distance to ball Mov by fitting a Gaussian to the distances of all previous "final" reinforcement signals.

If the type of the new reinforcement signal is "distance", then we expect that the robot has moved closer to ball Mov or ball WRT, depending on whether the robot is holding ball Mov or not.

In all other cases the distance from a ball could be anything, although it should be within a reasonable deviation from the distance when the last reinforcement signal was received.

if $(T_i = final)$
  $p(\vec{D}_i(j)|\ldots)$ is Gaussian with $\mu = \sum_K \vec{D}_k(j)$, and $\sigma = \sqrt{\frac{1}{n+1}\sum_k (\vec{D}_k(j) - \mu)^2}$, with k such that $k < i$ and $T_k = final$
else if $(T_i = distance) \cap ((Mov = j \cap H_i \neq j) \cup (H_i = Mov \cap WRT = j))$
  $p(\vec{D}_i(j)|\ldots)$ is Gaussian with $\mu = max(\vec{D}_{i-1}(j) - 30, 0)$ and $\sigma = 30mm$
else
  $p(\vec{D}_i(j)|\ldots)$ is Gaussian with $\mu = \vec{D}_{i-1}(j)$ and $\sigma = 500mm$

Note: $\vec{D}_i(j)$ takes on only positive values, while the domain of a Gaussian random variable is all real numbers. To account for this in the above equations we normalize the distribution by the area bellow zero.

3) $p(H_i|Mov, WRT, T^i, \vec{\Theta}^{i-1}, \vec{D}^{i-1}, H^{i-1})$*:* If the type of the new reinforcement signal is "final" or the robot was holding ball Mov when the previous reinforcement signal was received, then the robot must be holding ball Mov. That is, once the teacher has reinforced the robot for holding ball Mov we do not expect them to reinforce when the robot is not holding Mov, and the robot must be holding ball Mov if the teacher gave a "final" reinforcement.

In all other cases the probability of holding ball Mov should increase as the robot approaches ball Mov, and the probability of holding itself, ball 0, should decrease.

if $(T_i = final) \cup (H_{i-1} = Mov)$
  $p(H_i = Mov|\ldots) = 1$
  $p(H_i \neq Mov|\ldots) = 0$
else
  $p(H_i = Mov|\ldots) = f(\vec{D}_{i-1}(Mov))$
  $p(H_i = 0|\ldots) = 1 - f(\vec{D}_{i-1}(Mov))$
  $p(H_i \neq (Mov \cup 0)) = 0$

where $f(d) = max\left(\frac{p_{min} - p_{max}}{span} \times d + p_{max}, p_{min}\right)$. $p_{min} = 0.001, p_{max} = 0.8, span = 500mm$.

G. Motion Models

In our formulation of this problem T is the only hidden random variable that depends on time, and thus, T is the only hidden variable that needs a motion model.

1) $p(T_i|Mov, WRT, T^{i-1}, \vec{\Theta}^{i-1}, \vec{D}^{i-1}, H^{i-1})$*:* Once a reinforcement of type "final" is received, we assume that all subsequent reinforcements are also of type "final". If none of the reinforcement signals received so far have been of type "final", then the probability of the new reinforcement having type "final" increases as the robot approaches ball WRT, accounting for the distance it would takes to go and get ball Mov if it is not being held. The probability of receiving a reinforcement signal of type "attention" is equal to the probability of receiving a reinforcement signal of type "distance".

if $(T_{i-1} = final)$
  $p(T_i = final|\ldots) = 1$
  $p(T_i = attention|\ldots) = 0$
  $p(T_i = distance|\ldots) = 0$
else if $(H_{i-1} = Mov)$
  $p(T_i = final|\ldots) = f(\vec{D}_{i-1}(WRT))$
  $p(T_i = attention|\ldots) = \frac{1}{2}(1 - f(\vec{D}_{i-1}(WRT)))$
  $p(T_i = distance|\ldots) = \frac{1}{2}(1 - f(\vec{D}_{i-1}(WRT)))$
else
  $p(T_i = final|\ldots) = f(\vec{D}_{i-1}(Mov) + dist_{i-1}(Mov, WRT))$
  $p(T_i = attention|\ldots) = \frac{1}{2}(1 - f(\vec{D}_{i-1}(Mov) + dist_{i-1}(Mov, WRT)))$
  $p(T_i = distance|\ldots) = \frac{1}{2}(1 - f(\vec{D}_{i-1}(Mov) + dist_{i-1}(Mov, WRT)))$

where $f(d) = max\left(\frac{p_{min}-p_{max}}{span} \times d + p_{max}, p_{min}\right)$. $p_{min} = 0.001, p_{max} = 0.6, span = 2000mm$. $dist_i(b1, b2)$ is the distance between ball $b1$ and ball $b2$ when the ith reinforcement signal was received.

H. Prior Distribution at $i = 0$

At time zero, before any reinforcement signals have been received, the only hidden random variables are Mov and WRT. So the prior distribution needs to specify a probability for each assignment to the pair (Mov,WRT). We assume that all combinations are equally likely, i.e. before teaching, no ball is more likely to be moved than another and no ball is more likely to be moved with respect to than another. Thus, for all nine combinations of k and l, which excludes moving a ball to itself,

$$P(Mov{=}k, WRT{=}l) = 1/9. \tag{6}$$

I. Posterior Update

For this demonstration we chose to maintain the full posterior distribution, alternatives are described in the discussion section. Since all of our hidden random variables are discrete, the posterior distribution can be stored as a vector of probabilities, where each entry corresponds to one assignment to the hidden random variables. Before any reinforcement signals have been received, there are only two hidden random variables, Mov and WRT, with the nine possible assignments mentioned above. The posterior distribution begins as a vector with 9 elements all initialized to 1/9. After n reinforcement signals have been received, the posterior distribution is a vector with 9×3^n elements, since another T is added for each signal, and each T can take on one of three values, "final", "attention", or "distance". The nth signal is incorporated by creating a vector of length 9×3^n with three identical copies of the posterior vector from the last update, one corresponding to $T_n = final$, one for $T_n = attention$ and one for $T_n = distance$. Then, we follow Eq. (5) from back to front, by taking each element of the vector and multiplying by the motion model for that assignment to T_n, then multiplying by the measurement models for the observed H_n, $\vec{D}_n$, and $\vec{\Theta}_n$. This vector is then normalized to get the new posterior distribution.

More advanced methods do exist for computing posterior distributions [19], [20], [21]; the purpose of this paper is not to present a novel Bayesian computation algorithm, but to establish the basis of Bayesian inference as a method for learning task specifications from familiar human signals.

J. Action Selection

In this demonstration, the robot acts autonomously during the teaching process. In this section we describe the computation done to select a new "action".

As described above, the posterior distribution is maintained in a vector, where each element specifies the probability of an assignment to Mov, WRT, and T^i. The robot first uses this probability vector to sample an assignment of Mov, WRT, and T^i.

The robot could carry out the specified task to completion, but this is unnecessary. Because we assume the human teacher is "shaping" the task, the robot only needs to perform the task to the point where it expects to receive the next reinforcement. The distributions over the details of this next reinforcement signal are specified by the measurement and motion models. Thus, the robot chooses actions by sampling T_{i+1}, H_{i+1}, $\vec{D}_{i+1}$, and $\vec{\Theta}_{i+1}$, in that order (see Eq. (5)), conditioned on the posterior sample of Mov, WRT, and T^i, and then moving the world to that state. For example, if the value of the samples are ($Mov = 1$, $WRT = 2$, $H_{i+1} = 1$, $T_{i+1} = attention$, ...), then the robot should go pick up ball 1, because $H_{i+1} = 1$, and turn to face ball 2, because $T_{i+1} = attention$ and $WRT = 2$.

This procedure is run any time a reinforcement signal is received, or after three seconds of waiting once the state specified by the last sample has been reached.

VII. RESULTS

In the following results, the robot was taught by a non-technical graduate student. The results obtained appear to be typical of non-technical teachers. Similar results were obtained from six other non-technical teachers, with a minimum *teaching time* of roughly three minutes and a maximum *teaching time* of roughly seven minutes. Where *teaching time* was the time it took for the teacher to feel satisfied that the robot "knew" the task.

Fig. 3 shows the position of the robot, the three balls, and the reinforcement signals during the five minute teaching session[2]. The squares show the locations where reinforcement signals were received. Nine reinforcement signals were issued by the teacher. Roughly, the first four reinforcement signals showed the robot that it should move ball 1, the next three that it should move ball 1 to ball 2, and the final two refined the distance that ball 1 should be from ball 2.

Fig. 4 shows the robot performing the learned task. The robot captures ball 1 and takes it directly to a distance of one foot from ball 2.

In Fig. 5 we plot the distance from the robot to each of the three balls vs. time for the same teaching and execution runs. The vertical lines mark the times when a reinforcement signal was received. During the teaching session the distance to ball 1 drops first, since the robot is shown that ball 1 is to be moved, followed by a drop in the distance to ball 2, as the robot is taught that ball 1 should be moved to ball 2.

Fig. 6 illustrates the convergence of Mov and WRT to the desired values. The left column shows the distribution's for Mov and WRT after each of the nine reinforcements. $P(Mov = 1)$ and $P(WRT = 2)$ both converge to 1. The right column shows the entropy for Mov and WRT.

$$Entropy(X) = -\sum_{x_i} p(x_i) \times log(p(x_i)). \tag{7}$$

[2]Visit *http://eecs.harvard.edu/∼woodward/videos* for the video of this teaching session.

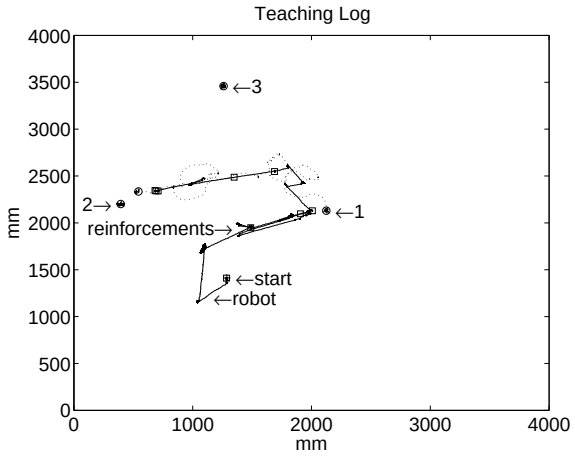

Fig. 3. This figure shows the actual trajectory of the robot and balls, and the reinforcement locations during the teaching session. The "robot" line shows the path of the robot, which starts at (1300mm, 1400mm). The squares indicate the locations where the reinforcement signals were received. The circles show the start and stop positions of the three balls, with the path in between drawn as a dotted line; only ball 1 moved during this teaching session. Nine reinforcement signals were issued. The last two overlap the 7th signal. The robot operated autonomously during the teaching session. This task was taught in under 5 minutes. Similar results were obtained from six other non-technical users.

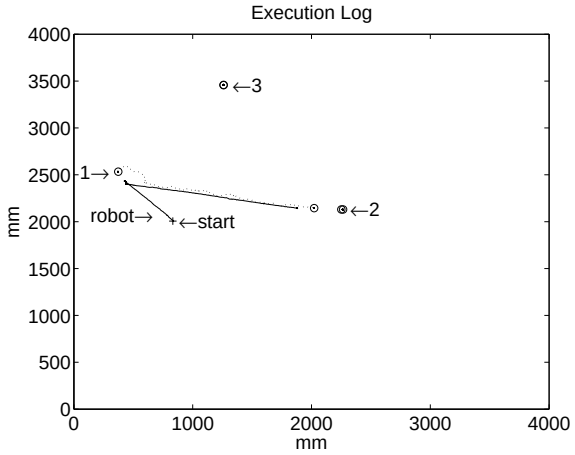

Fig. 4. Once the task has been taught to the robot, the robot can accurately and repeatedly execute the task. In future work the teacher will be able to associate a trigger with this task so that it, and other tasks, can be executed on command.

We use entropy as a measure of the uncertainty in the two random variables. Both variables start at complete uncertainty and converge to near certainty that $Mov = 1$ and $WRT = 2$.

VIII. DISCUSSION

A. Measurement and Motion Model design

In the demonstration above, the form and parameters of the models were set based on intuition. One approach to more accurate models would be to survey humans as they teach tasks to the robot or another human, and then fit the models to the survey data.

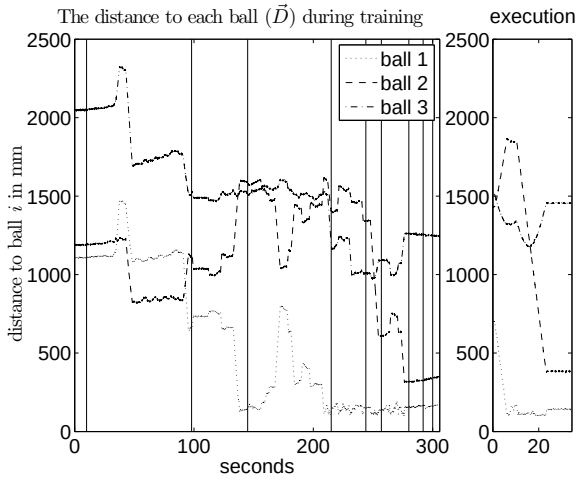

Fig. 5. This figure shows another view of the teaching session and execution run. The distance from the robot to each of the three balls is plotted vs. time for the teaching session (left) and execution run (right). The vertical lines show when each of the nine reinforcements were received. The distance to ball 1 drops first, followed by the distance to ball 2, since the robot is first shown that ball 1 is to be moved, and then that it is to be moved to ball 2.

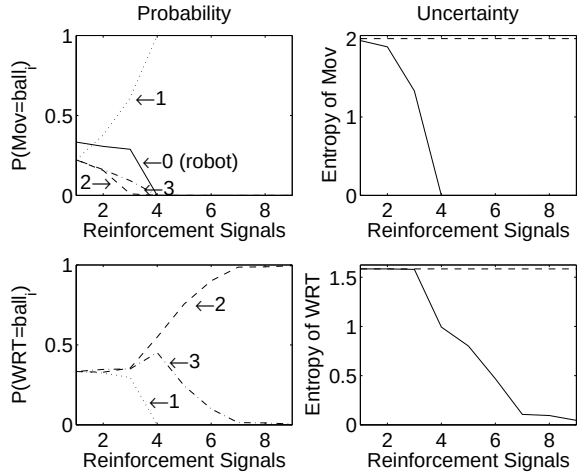

Fig. 6. This figure illustrates the accuracy that Bayesian Inference brings to the problem of teaching high-level tasks. When a robot perceives a teaching signal from a human, such as the reinforcement "spacebar" press in our example, there is nearly always ambiguity regarding the intentions of the signal. For example with the reinforcement "spacebar" press the robot knows that it is doing something that the teacher approves of, but was it that the robot was "looking" at the green ball, or was it that the robot was near the red ball? An incorrect guess could lead the robot to the wrong conclusion. Through Bayesian inference the robot can maintain all possibilities, and gradually arrive at the correct conclusion as more evidence arrives. The top row shows the probability and uncertainty of which ball is to be moved (Mov) and the bottom row shows the probability and uncertainty for which ball Mov is supposed to be moved with respect to (WRT). We use Entropy as the measure of uncertainty. The dashed line in the uncertainty column shows complete uncertainty for that random variable. As more reinforcement signals arrive the uncertainty about which object to move and which object to move it with respect approaches zero.

B. Exponential Growth of the Posterior Space

In our application of Bayesian inference for the demonstration we maintained a vector of probabilities, one for every point in the posterior space, i.e. every assignment to the tuple

of hidden random variables. The problem with computing probabilities for every point is that the number of points grows exponentially with each new measurement, and thus updating a probability for each point could quickly consume more memory and processing cycles then are available. In many cases, the complete posterior distribution is not needed, only samples from that distribution. Metropolis-Hastings and particle filtering are two techniques for drawing such samples [20], [19]. Particle filtering seems most promising since, unlike Metropolis-Hastings, it does not require a measure of distance in the posterior space and a meaningful measure of distance is hard to specify when discrete random variables are used.

C. Evaluation Metrics and Comparisons

In this paper we proposed the application of Bayesian inference to task learning and demonstrated one possible implementation. As new techniques are developed to deal with larger task complexity and additional signals, we will need metrics for evaluating their effectiveness. Here are some possible metrics: the time to teach a task, the time for the teacher to learn the system, and how enjoyable the teacher found the experience.

These metrics could also be used to compare Bayesian inference to another "learning from a human teacher" approach, or even to a human learning the same task.

D. Expansion of Demonstration

In our demonstration of Bayesian inference for task learning, the type of task the robot could be taught to perform was very simple, namely, to move one object to a distance from another object. Some of the immediate extensions we are planning are: tasks involving everyday objects, the learning of multiple tasks, the association of triggers with tasks, and tasks that involve multiple steps (chaining).

IX. CONCLUSION

We have proposed the use of Bayesian inference to process signals from a human teacher and learn a task, which we summarize here. To apply Bayesian inference the following must be specified: hidden random variables, observable random variables, measurement models, motion models, and a prior distribution. In the case of learning tasks from humans, the hidden random variables capture the task specification, the observed random variables capture measurements from the human signals. Measurement models predict the signals, motion models describe evolving characteristics of the human teacher, and the prior distribution captures the belief about the possible tasks *before* teaching.

We then demonstrated this approach on a robot capable of moving simple objects around a room. A human teacher pressed the spacebar to administer a reinforcement signal, taken to mean that the teacher approved of something the robot was doing. After five minutes and nine reinforcements the robot learned the details of the task and was able to repeat the task once taught.

The use of Bayesian inference will allow robots to make use of any signal the teacher may use, however ambiguous, to learn a task.

REFERENCES

[1] A. Saxena, J. Driemeyer, J. Kearns, and A. Y. Ng, "Robotic grasping of novel objects," in *Proc. of the Twenty-First Annual Conference on Neural Information Processing Systems (NIPS)*, 2007.

[2] E. Klingbeil, A. Saxena, and A. Y. Ng, "Learning to open doors," in *The 17th Annual AAAI robot workshop and Exhibition*, 2008.

[3] S. Thrun, M. Montemerlo, H. Dahlkamp, D. Stavens, A. Aron, J. Diebel, P. Fong, J. Gale, M. Halpenny, G. Hoffmann, K. Lau, C. Oakley, M. Palatucci, V. Pratt, P. Stang, S. Strohband, C. Dupont, L.-E. Jendrossek, C. Koelen, C. Markey, C. Rummel, J. van Niekerk, E. Jensen, P. Alessandrini, G. Bradski, B. Davies, S. Ettinger, A. Kaehler, A. Nefian, and P. Mahoney, "Winning the darpa grand challenge," *Journal of Field Robotics*, 2006.

[4] P. Abbeel and A. Y. Ng, "Apprenticeship learning via inverse reinforcement learning," in *Proc. of the Twenty-first International Conference on Machine learning (ICML)*, 2004.

[5] W. D. Smart and L. P. Kaelbling, "Effective reinforcement learning for mobile robots," in *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, 2002.

[6] M. Nicolescu, O. C. Jenkins, and A. Olenderski, "Learning behavior fusion estimation from demonstration," in *Proc. IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, Sept. 2006, pp. 340–345.

[7] M. Nicolescu, O. C. Jenkins, and A. Stanhope, "Fusing robot behaviors for human-level tasks," in *Proc. IEEE International Conference on Development and Learning (ICDL)*, July 2007, pp. 76–81.

[8] O. Lebeltel, P. Bessiere, J. Diard, and E. Mazer, "Bayesian robots programming," in *Research Report 1, Les Cahiers du Laboratoire Leibniz, Grenoble (FR)*, 2000, pp. 49–70.

[9] A. Levas and M. Selfridge, "A user-friendly high-level robot teaching system," in *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, Mar. 1984, pp. 413–416.

[10] F. Kaplan, P. Y. Oudeyer, and E. Kubinyi, "Robotic clicker training," *Robotics and Autonomous Systems*, vol. 38(3-4), pp. 197–206, 2002.

[11] B. Blumberg, M. Downie, Y. Ivanov, M. Berlin, M. P. Johnson, and B. Tomlinson, "Integrated learning for interactive synthetic characters," in *Proc. International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, 2002.

[12] M. N. Nicolescu and M. J. Mataric, "Natural methods for robot task learning: Instructive demonstration, generalization and practice," in *Proc. of the Second International Joint Conference on Autonomous Agents and Multi-Agent Systems*, July 2003, pp. 241–248.

[13] A. Lockerd and C. Breazeal, "Tutelage and socially guided robot learning," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, vol. 4, Sept. 2004, pp. 3475–3480.

[14] D. S. Touretzky and L. M. Saksida, "Operant conditioning in skinnerbots," *Adaptive Behavior*, vol. 5, p. 34, 1997.

[15] L. M. Saksida, S. M. Raymond, and D. S. Touretzky, "Shaping robot behavior using principles from instrumental conditioning," *Robotics and Autonomous Systems*, vol. 22(3/4), p. 231, 1998.

[16] A. L. Thomaz, G. Hoffman, and C. Breazeal, "Reinforcement learning with human teachers: Understanding how people want to teach robots," in *Proc. IEEE International Symposium on Robot and Human Interactive Communication (ROMAN)*, Sept. 2006, pp. 352–357.

[17] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA: MIT Press, 1998.

[18] C. Breazeal, "Emotion and sociable humanoid robots," *International Journal of Human Computer Interaction*, vol. 59, pp. 119–155, 2003.

[19] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. Cambridge, MA: MIT Press, 2005.

[20] A. Gelman, J. B. Carlin, H. S. Stern, and D. B. Rubin, *Bayesian Data Analysis*, 2nd ed. Boca Raton, FL: Chapman & Hall/CRC, 2004.

[21] A. Gelman and J. Hill, *Data Analysis Using Regression and Multilevel/Hierarchical Models*. New York, NY: Cambridge University Press, 2007.