Rapid Behavior Adaptation For Human Centered Robots Through Demonstration

Saifuddin Md. Tareeq

DOCTOR OF PHILOSOPHY

Department of Informatics
School of Multidisciplinary Sciences
The Graduate University for Advanced Studies

2010

A dissertation submitted to the Department of Informatics,
School of Multidisciplinary Sciences,
The Graduate University for Advanced Studies (SOKENDAI)
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy

Advisory Committee

Assoc. Prof. Tetsunari Inamura	National Institute of Informatics, SOKENDAI
Prof. Seiji Yamada	National Institute of Informatics, SOKENDAI
Prof. Ken Satoh	National Institute of Informatics, SOKENDAI
Prof. Haruki Ueno	National Institute of Informatics, SOKENDAI
Assoc. Prof. Ryutaro Ichise	National Institute of Informatics, SOKENDAI

Tareeq, Saifuddin Md. (Ph.D.)

Rapid Behavior Adaptation for Human Centered Robots Through Demonstration

Thesis directed by Assoc. Prof. Tetsunari Inamura

The problem of learning behavior policy, a task representation mapping from world states to actions, lies at the heart of many robotic applications. One approach to acquiring behavior policy is learning from demonstration, an interactive technique in which a robot learns a policy based on observation to action mappings provided by a human teacher. When the user behavior policy changes frequently the robot has to adapt to new user behavior policy rapidly. Robots ability to rapidly adapt to user behavior policy is an important aspect of learning from observation because otherwise the user may be tired. However rapid adaptation poses a significant challenge because it is difficult with conventional methods. Bayesian network is suitable to address the challenge because it can represent degree of confidence for behavior decision as probability and can provide a confidence even with a small number of observations. Bayesian network is also suitable for online interactive learning. However the performance of Bayesian learning strongly depends on the quality of the observed dataset. When the dataset included significant data, the learning would be a success. But it is difficult to evaluate data to be insignificant because when the data become insignificant for learning process is not known a priori. In this thesis, we propose a novel concept for evaluating significance of data and contributes multiple algorithms that provide rapid behavior adaptation. We evaluate our algorithms empirically, both in simulation and on a real robot. Results with the performance PeopleBot confirm the effectiveness of the algorithms in rapid behavior adaptation.

Dedication

To my parents.

Acknowledgements

There are many people who helped me produce this thesis document. Without the help of all of my friends, family, and colleagues, this thesis document would have never been completed.

I would first like to express my sincere gratitude to my supervisor Tetsunari Inamura for his constructive criticism, guidance, support and for sharing his wealth of knowledge, all of which have had a strong influence on this thesis. I would also like to thank him for providing me the appropriate environment for research, and for being accessible at all times.

I would like to thank all the members of Intelligent Interactive Robotics Lab over the years. Without the experiences we shared together, I would have never known to even investigate the problems addressed in this thesis. I would also especially like to thank Keisuke Okuno for listening to me, keeping me on track, and just generally being there whenever I needed a friend.

I would like to thank my family for having faith in me to finish. They have never doubted me and their confidence helped in rough times.

Contents

Chapter

Appendix

Tables

Table

Figures

Figure

Chapter 1

Introduction

Robots have proven powerful tools in the predictable environments of factories and manufacturing plants. However, they have been far less successful in human environments characterized by a higher degree of uncertainty and change. Each response of today's industrial robots has to be programmed in advance. This approach is ill suited for robots in human environments, which require a vast amount of knowledge and the specification of a wide set of behaviors for successful performance. Typically robots in human environments are placed in very restricted worlds because then the environment can be controlled. If a robot is taken in a unknown home, that approach just doesn't hold anymore. Moreover when the user or environment changes frequently the robotic system should be able to adapt to new user or environment rapidly to take correct action. This has introduced the need for building robotic systems able to adapting to user and environment in an engaging way by using their observed sensory information.

The recent trend in robotics is to develop a new generation of robots that are capable of adapting to new user, interacting with user and participate in our daily lives. Adaptive behavior plays an important role in the assistance of different user with different needs. Therefore, such robots should be able to rapidly adapt to user preference, user policy and have interaction skills to communicate with user. In this thesis users preference indicates variation of behavior decision by the user even though identical sensor is observed. And user's policy is defined by the mapping from observation to action. The problem of learning a policy, a task representation mapping from world states to actions, lies at the heart of many robotic applications. One approach to acquiring a task policy is learning from demonstration, an interactive learning approach based on human-robot interaction that provides an intuitive interface for robot programming. In this approach, a teacher performs demonstrations of the desired behavior to the robot. The robot records the demonstrations, typically as state to action mappings, and learns a policy imitating the teachers behavior.

Learning from demonstration is an incremental online learning process in which the robot begins with no knowledge about the task, and acquires training data until a fully autonomous policy representing the complete task is learned. If the user change his preference or policy the system should adapt to the new preference or policy rapidly. This thesis contributes an interactive approach to demonstration learning that enables the robot to rapidly adapt to user preference and policy. In this thesis users preference indicates variation of behavior decision by the user even though identical sensor is observed and policy is the mapping from world state to action. These algorithms enables the robot to identify the need for and request demonstrations for specific parts of the state space based on confidence thresholds characterizing the uncertainty of the learned policy. In our evaluation, we show that this approach significantly reduces the number of demonstrations and can rapidly follow user preference and policy.

1.1 Rapid policy adaptation

Demonstrations provide the robot with a dataset consisting of state-action pairs representing examples of the desired behavior. The robots goal is to use this information to adapt to a policy, which enables the robot to select an action based upon its current world state. Our policy should map from the robots state to a discrete set of action primitives. And due to the interactive nature of learning from demonstration, policy adaptation must occur rapidly. In this thesis rapidness is defined as the ratio of expected time to change user's preference or policy by the user to the required time to adapt to preference or policy by the robot.

The state-action mapping represented by a policy is typically complex. One reason for this complexity is that the desired observation-action mapping is unknown. A second reason for this complexity is the complications of policy adaptation in real world environments. Traditional approaches to robot control model the domain dynamics and derive policies using mathematical models. Though theoretically well-founded, these approaches depend heavily upon the accuracy of the model. Not only does this model require considerable expertise to develop, but approximations such as linearization are often introduced for computational tractability, thereby degrading performance. Other approaches, such as reinforcement learning, guide policy learning by providing reward feedback about the desirability of visiting particular states. To define a function to provide these rewards, however, is known to be a difficult problem that also requires considerable expertise to address. Furthermore, building the policy requires gathering information by visiting states to receive rewards, which is non-trivial for a mobile robot learner executing actual actions in the real world. We chose Bayesian network for rapid policy adaptation because it can represent degree of confidence for behavior decision as probability and can provide a confidence even with a small number of observations. Also Bayesian network is suitable for online interactive learning.

1.2 Approach

This thesis presents a Bayesian network based framework to address rapid behavior adaptation. The performance of Bayesian learning strongly depends on the quality of the demonstration dataset. When the dataset included significant data, the learning would be a success. But it is difficult to evaluate data to be insignificant because when the data become insignificant for learning process is not known a priori. We propose a method for evaluating significance of data based on a concept of change in the degree of confidence. A small change in the degree of confidence can be regarded as an insignificant data for learning, so that data will be evaluated as insignificant.

For evaluating the significance of demonstration, the experience data is assigned to distribution parameters. The distribution represents not only event probability among behaviors, but also degree of confidence for the output probability. The system calculates the degree of confidence by integrating the area around peak of the distribution after each demonstration. The change in the two consecutive degrees of confidence can be regarded as the importance of the observation to the learning process. When the change in the degree of confidence in two consecutive time steps is small, this situation is regarded as familiar; the experience data is considered insignificant for learning and be discarded. In contrast, when the robot detect a large change in the degree of confidence in two consecutive time steps, this situation is considered unfamiliar; the experience data is considered significant for learning and be accepted.

We introduce multiple rapid behavior adaptation algorithms that enable the robot to evaluate demonstrations based on the change in the degree of confidence. The rapid adaptation algorithm enables the robot to evaluate demonstrations in real time as it interacts with the user. To enable the robot to clarify unfamiliar and ambiguous situation, we assume that the robot can pause between executions of consecutive actions. The clarification component allows the teacher to provide demonstration when the confidence of the robot is low. Combined, these techniques provide a rapid and intuitive approach for rapid behavior adaptation.

We utilize the learning from experience approach, which has the advantage that the demonstrations are inherently restricted to the robots physical abilities and a mapping from the teachers actions to those of the robot is not required. The teacher does not perform demonstrations with his body. Instead the teacher to select among the robots available actions. We assume the robot has a discrete number of of high-level actions, or action primitives, which are the basic building blocks required to perform the task. This ensures that the teachers demonstrations always fall within the robots abilities.

1.3 Contributions

The main contributions of the thesis are:

A novel method for rapid preference adaptation. We show that the method of significance evaluation can be used to handle prior probability for rapid preference adaptation. Change in degree of confidence based significance evaluation method is used to select important action so that prior information become uninformative. It made rapid preference adaption possible. We demonstrate the utility of rapid preference adaptation method in the mobile robot context. We show that robot can automatically adapt rapidly to user preference when significance evaluation is used. This work is published in conferences (reviewed) [103], [105] and is accepted in journal [106].

A novel method for rapid policy adaptation. We show that the method of significance evaluation can be used for rapid policy adaptation. Significance evaluation method is used to select important observation for addressing both prior and conditional probability. We show that robot can automatically adapt rapidly to user policies when significance evaluation is used. This work is published in conferences (reviewed) [107] and in journal [108].

Two techniques for representing and teaching collaborative behavior using demonstration. These techniques are based on different information sharing strategies: implicit coordination and coordination through active communication. We are now working on interactive communication that can be used for handling unstable confidence for rapid policy adaptation. Preliminary result of this work is presented in the conferences [104].

In current technology, there is no system that allows a robot to adapt in real-time to user's policy. Therefore, the aim of this thesis is to find a solution to this problem by proposing a system by which such a rapid adaptation is possible.

1.4 Organization of the thesis

The work of this thesis is organized in the following chapters.

Chapter 2-Related Works we discuss the related works to this thesis.In this chapter, We start off with an overview of the different viewpoints from which the problem of policy adaptation has been approached. We then describe the related work and their limitations in relation to rapid behavior adaptation. We finish by highlighting the reasons of our choice for Bayesian network.

Chapter 3- Rapid adaptation to user preference starts with a brief description of Bayesian network. We then describe the problem in case of Bayesian network. We then describe the concept of change in the degree of confidence and how to take the advantage with beta and Dirichlet distribution. Then we describe how user preference can be followed with the change in degree of confidence based significance evaluation. Experiment and result for rapid adaptation to user preference is also described.

Chapter 4- Rapid adaptation to user policy describe how change in degree of confidence based significance evaluation can be applied for policy adaptation. We described the the experiment and result for user policy adaptation in this chapter.

Chapter 5-Conclusion and future work discusses the contribution of the thesis. Give limitations of the thesis and provide future directions. As a future work we include interactive communication so that the robot can make interaction with the user by asking question or clarifying situation when it has low confidence while adapting to new policy.

Chapter 2

Related work

2.1 Introduction

Algorithms for rapid adaptation will become an important prerequisite for future robots to achieve a more intelligent coordination of their movements that is closer to human. In this thesis we focus on demonstration based rapid behavior adaptation. Learning from demonstration is a technique to derive a policy by a learner from given examples of behavior. There are several common aspects of demonstration based learning among all applications to date. One is the fact that a teacher demonstrates execution of a desired behavior. Another is that the learner derives a policy to reproduce the demonstrated behavior with a set of these provided demonstrations. Considering the two fundamental phases we segment the demonstration learning problem into data gathering and policy learning and survey different solutions to each in this chapter. We also discuss other adaptation techniques than demonstration based adaptation. We discuss about the limitation of existing methods and clarify why we chose Bayesian network as our framework.

2.2 Data Gathering

This section discusses demonstration execution and recording techniques. The dataset is recorded during teacher executions of a desired behavior and is composed of state-action pairs. Tools used for behavior execution and procedure used for recording varies greatly across approaches. Examples range from sensors on the robot learner recording its own actions as it is passively teleoperated by the teacher, to a camera recording a human teacher as she executes the behavior with his own body.

2.2.1 Learning from experience

In learning from experience data gathering approach, the robot experience the demonstrated actions using its own body. This demonstration experience is gained by the robot through exploitation of its parts or by instructing it which of its actions to perform. The robot is an integral part of the demonstration and executes the actions selected by the teacher while using it's sensors to sense the environment. This approach eliminates the correspondence problem faced by observation-based methods by allowing the robot to directly associate gathered sensory information with one of its own actions. There may exist an indirect record mapping for state-action pairs when teacher execution is demonstrated which needs to be inferred from the data.

2.2.1.1 Method to provide demonstration

We identify two methods for providing demonstration data to the robot learner based on the record mapping distinction:

(1) *Teleoperation*: A demonstration technique in which the robot's sensors record the execution while the robot is operated by the teacher.

(2) *Teacher following*: A demonstration technique in which the robot attempts to match or mimic the teacher's motion as the teacher executes the task while the robot learner records the execution using its own sensors.

During teleoperation, a robot records from its own sensors while it is operated by the teacher. Within demonstration learning, teleoperation provides the most direct method for information transfer. However, teleoperation requires that operating the robot be manageable. This requirement limits the suitability of this technique to some systems. For example low-level motion demonstrations are difficult on systems with complex motor control, such as high degree of freedom humanoids. The positive side of the teleoperation approach is the direct transfer of information from teacher to learner, while its negative side is the requirement that the robot be operated in order to provide a demonstration.

Demonstrations recorded through human teleoperation via a joystick are used in a variety of applications, including obstacle avoidance and navigation [46][94],robot kicking motions [19], flying a robotic helicopter [77], object grasping [84][98] and robotic arm assembly tasks[22]. Teleoperation is also applied to a wide variety of simulated domains, ranging from static mazes [28][87] to dynamic driving [1] [25] and soccer domains [3] and many other applications. In place of a human teacher, hand-written controllers are also used to teleoperate robots [42] [88][94].

During teacher following, the robot records from its own sensors while trying to mimic the teacher's demonstrated motions. The states or actions of the true demonstration execution are not recorded. Instead the learner records its own mimicking execution and thus indirectly encode the teacher's states/actions within the dataset. An extra algorithmic component is required in teacher following in comparison to teleoperation that enables the robot to track and actively follow (rather than passively be teleoperated by) the teacher. Indeed the teacher following does not require that the teacher be able to operate the robot in order to provide a demonstration.

Demonstrations recorded through teacher following teach navigational tasks by having a robot follow an identical-platform of robot teacher through a maze [32], follow a human teacher past sequences of colored markers [79] and mimic routes executed by a human teacher [76]. Teacher following also has a humanoid learn arm gestures, by mimicking the motions of a human demonstrator [80].

It is important to note that sometimes there exist a significant gap between the full observation state of the teacher and the demonstration data recorded by real robots. This occurs when the teacher employs extra sensors that are not recorded while executing demonstration. For example if there exist some inaccessible parts of the world to the robot's camera while the teacher can observe it (e.g. behind the robot, if its cameras are forward-facing), then the state differs from what is actually observed by the teacher and recorded as data. Teleoperation requires that operating the robot be manageable,and as a result not all systems might be suitable for this technique. Teacher following requires that the learner be able to identify and track the teacher but has the advantage of not requiring the teacher to actively control the robot. Furthermore, no direct recording of the observation is made by the teacher during the execution.

2.2.2 Learning from observation

Demonstration examples are obtained by the robot by passively observing the actions of the teacher in this learning from observation data gathering approach. It is assumed in learning from observation method that the task being learned is best demonstrated by the teacher through independent and uninhibited execution using his own body. This approach is commonly used to demonstrate low-level motion control of the robot for which teleoperation would be difficult such as high degree of freedom robots (humanoids). Common applications include object manipulation [62] [115] and interactive games [52] [10] [9].

Interpretation and extraction of useful information from the observed demonstration is done by the robot in learning from observation data gathering approach. This gives rise to the correspondence problem [31], the challenge of identifying the state and action of the demonstrator and mapping them to the robots own abilities with the corresponding effect. Accurate and efficient methods for solving this problem have been the focus of extensive research [4] [31] [45] [54] [55] [115].

To record the demonstration and recognize the actions performed by the demonstrator, most observation-based methods rely on advanced computer vision systems [12][54] [55]. Extracting meaningful information from image sequences is a very challenging problem, and marker-based vision systems have been developed that provide additional information about the demonstrators movements. Additional data sources have included tactile [33] [115], position [22] [45], and magnetic sensors [33], as well as speech recognition [33] [64] [90].

2.2.3 Interactive communication

In interactive demonstration, data is acquired through an active communication between robot and teacher. In this approach the robot acts as a collaborator by providing feedback to the teacher about the learning process. In this approach the robot receive demonstrated examples actively and can indicate uncertainty or even asking questions about the task. This approach allows the robot to help guide the learning process. This approach builds upon interaction framework and can be considered as a natural extension of demonstration learning. It is important to note that interactive demonstration is not mutually exclusive from the previous two data gathering techniques. Interactive communication based demonstration describes the method by which demonstrations are selected, and not the way they are performed. To learn through interaction with human teachers and other autonomous robots a number of algorithms have been developed. In the context of reinforcement learning, Clouse presents the *ask for help* framework [28]. This approach enables a robot to request advice from other robots when it is confused about what action to take. This confusion is described by relatively equal quality estimates for all possible actions in a given state.

Chernova *et. al.* [26][24] introduces confidence-based autonomy, a mixed-initiative robot demonstration learning algorithm that enables the robot and teacher to jointly control the learning process. The robot identifies the need for and requests demonstrations for specific parts of the state space based on confidence thresholds characterizing the uncertainty of the learned policy. Nicolescu [78] present a learning framework based on demonstration, generalization and teacher feedback, in which training is performed by having the robot follow a human and observe its actions. A high-level task representation is then constructed by analyzing the experience with respect to the robots underlying capabilities. The authors also describe a generalization of the framework that allows the robot to interactively request help from a human in order to resolve problems and unexpected situations. This interaction is implicit as the robot has no direct method of communication; instead, it attempts to convey its intentions by communicating through its actions.

Argall *et al.* [6] [5] present methods for policy refinement and generation using feedback. Their algorithms use feedback to refine demonstrated policies, as well as to build new policies through the scaffolding of simple motion behaviors learned from demonstration. Lockerd and Breazeal [17] [64] demonstrate a robotic system where high-level tasks are taught through social interaction. In this framework, the teacher interacts with the robot through speech and visual inputs, and the learning robot expresses its internal state through emotive cues such as facial and body expressions to help guide the teaching process. The outcome of the learning is a goal-oriented hierarchical task model. In later work

[109], the authors examine ways in which people give feedback when engaged in an interactive teaching task. Although the studys focus is to examine the use of a human-controlled reward signal in reinforcement learning, the authors also find that users express a desire to guide or control the robot while teaching. This result supports our belief that, for many robotic domains, teleoperation provides an easy and intuitive human-robot communication method.

Grollman and Jenkins present the dogged learning algorithm [40], a confidence-based learning approach for teaching low-level robotic skills. In this algorithm, the robot indicates to the teacher its certainty in performing various elements of the task. The teacher may then choose to provide additional demonstrations based on this feedback. Inamura *et al.* [46] [47][48]present similarly motivated methods based on Bayesian Networks that are limited to a discretely-valued feature set.

In addition to learning from demonstration, within machine learning research, active learning [14][29] enables a learner to query an expert and obtain labels for unlabeled training examples. Aimed at domains in which a large quantity of data is available but labeling is expensive, active learning directs the expert to label the more informative examples with the goal of minimizing the number of queries.

2.3 Policy learning

Robot must learn a policy that enables it to reproduce the desired behavior Once the demonstration data has been obtained. Three core approaches to deriving policies from demonstration data has been developed in case of learning from demonstration. Learning a policy can involve learning an approximation of the state to action mapping (**mapping function**), or learning the model of the world dynamics and deriving a policy from this information (**system model**). Alternately, a sequence of actions can be produced by a planner after learning a model of action pre- and post-conditions (**plans**). Across all of these learning techniques, minimal parameter tuning and rapid learning times requiring few training examples are desirable.

2.3.1 Mapping Function

The mapping function approach to policy learning find a function that approximates the state to action mapping for the demonstrated behavior. The goal of this type of algorithms is to reproduce the underlying unknown teacher policy. And, if possible, generalize over the set of available training examples such that valid solutions are also acquired for similar states that may not have been encountered during demonstration. Many factors influence the details of function approximation. These include whether the state input and action output are continuous or discrete; whether the generalization technique uses the data directly at execution time or uses the data to approximate a function prior to execution time; whether it is feasible or desirable to keep the entire demonstration dataset around throughout learning; and whether the algorithm updates online.

In general, mapping approximation techniques fall into two categories depending on whether the prediction output of the algorithm is discrete or continuous. Classification techniques are used to produce discrete output. Example classification methods applied to learning from demonstration have included k-Nearest Neighbors [92], Bayesian Networks [17] and Hidden Markov Models [49][50][51][34]. Regression techniques produce continuous output. Regression algorithms applied to demonstration learning for robotic systems include Locally Weighted Regression [19][73][95][40], Gaussian Mixture Regression [20], and On-Line Gaussian Processes [42]. A common assumption held by these approaches is that for any world state there exists a single best action.

2.3.2 System Model

The system model approach to policy learning determines a state transition model for the world from which it then derives a policy. This approach is typically formulated within the structure of reinforcement learning [97]. The standard reinforcement learning system assumes that the world can be described by a finite set of states, and that the robot is limited to taking one of a finite set of actions at each discrete timestep. At each learning timestep, the robot observes its state in the world and chooses an action to execute. After completing the action, the robot receives a reinforcement signal, or reward, that reflects the goodness of the action or the resulting state. The robots goal is to learn a mapping from states to actions that maximizes the robots reward over time. Among the many extensions to the standard reinforcement learning algorithm, a wide range of approaches have been developed for applying reinforcement learning to continuous state and action domains [71] [72] [96] [97] [113].

Within the context of reinforcement learning, demonstration can be seen as a source of reliable information, or advice, that can be used to accelerate the learning process [65] [81] [88]. A number of approaches for taking advantage of this information have been developed, such as deriving or modifying the reward function based on the demonstration [1] [7] [28] [82] [109] and using the demonstration experiences to prime the robots value function or model [85] [93] [99]. [96] propose an alternate approach for accelerating learning in domains with sparse rewards. Instead of attempting to model expert behavior by rewarding similar actions, they allow the expert to demonstrate the execution of the task and expose the robot to areas of the state space where the reward is non-zero. Bootstrapped with this information, the robot is then able to learn the task under autonomous execution.

2.3.3 Plans

A policy is represented as a sequence of actions that lead from the initial state to the final goal state in the planning framework. Actions are often defined in terms of pre-conditions, the state that must be established before the action can be performed, and post-conditions, the state resulting from the actions execution. Unlike other learning from demonstration approaches, planning techniques frequently rely not only on state-action demonstrations, but also on additional information in the form of annotations or intentions from the teacher [37] [39] [63]. Demonstration-based algorithms differ in how the rules associating pre- and post-conditions with actions are learned, and whether additional information is provided by the teacher.

Veeraraghavan and Veloso [114] present an algorithm for learning generalized plans that represent sequential tasks with repetitions. In this framework, a humanoid robot is taught the repetitive task of collecting colored balls into a box based on two demonstrations. Rybski *et al.* [91] present a system in which demonstration of the desired task is performed through speech dialog. The teacher presents the robot with a series of conditional statements which are processed into a plan. The robot is additionally able to verify any unspecified parts of the plan through dialog.Nicolescu [78] introduce a plan-based framework in which training is performed by having the robot follow a human and observe its actions. A high level task representation is constructed by the algorithm by analyzing this experience with respect to the robots abilities.

2.4 Learning algorithms and their problem in rapid adaptation

Several machine learning approach is proposed for adaptation in the past that include Reinforcement Learning (RL) and Artificial Neural Network (ANN) obtaining successful result. Here we discuss related algorithms to identify their unsuitability for rapid adaptation.

Reinforcement Learning

Reinforce Learning (RL) [97] is a machine learning technique that has been frequently used in many domains. Although the obtained results are usually successful, the main drawback of this technique is the large state space that most problem present. As a consequence, a large amount of learning steps are required to find the policy that matches states and actions. Hence most of the times this technique is not feasible when dealing with real robots.

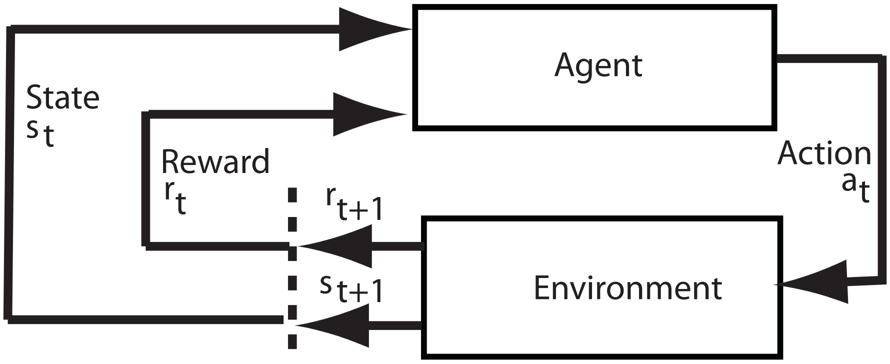


Figure 2.1: Reinforcement learning architecture. Adapted from [97]

In the field of reinforcement learning, almost all learning agents gain experience solely by interaction with their environmentteachers are not in the loop. Recently the idea of integrating a teacher into the learning process has been proposed [1] [116][28]. The field of Inverse Reinforcement Learning [1], also sometimes called apprenticeship learning, attempts to learn an agents reward function by observing a sequence of actions (not rewards) taken by a teacher. In contrast to this interaction, [116] consider a teacher with a policy that can deliver a trace to the learning agent after seeing it behaving sub-optimally. In that the learners actually see samples of the transitions and rewards collected by the teacher and use this 'experience' in a traditional RL fashion.

Kleiner *et. al.* [59] apply a hierarchical reinforcement learning in a semi Markov Decision Process framework. In their approach they show that learning simultaneously on the behavior level (low level) and on the policy level (high level) is advantageous with respect to only on one level at a time. Result with the real robots when policy is obtained through simulation show that more than an hour would be necessary to improve the hand-coded action selection mechanism.

Neural Network

Neural network [89] have been proved to efficiently perform in many domains, including robot control. However one of their main drawbacks, is the large amount of data needed for the training, which is not always feasible to provide. Another problem with conventional Artificial Neural Network (ANN) based methods is that the diversity of behavior that can be learned by a single conventional ANN is strongly limited by the degree to which a number of behavior systems can be realized in a single functional mapping. This limits the methods capacity for rapid adaptation to user policies.

Because feed-forward networks compute a one-way mapping from inputs to outputs (i.e., in the case of a robot controller from sensor input to motor outputs), a feed-forward controller can only react to its current input in each time step. The complexity of behavior that can be achieved with such an agent is, of course, limited. Nevertheless, it has been shown that such a system can learn to acquire far more than trivial behavior. Nehmzow [75], for example, used a simple robotic vehicle controlled by a feed-forward network mapping the input from two whisker sensors at the front of the vehicle to four possible actions (left, right, forward, backward). It was shown that, using reinforcement learning, this vehicle could successfully be trained to perform tasks like following corridors or pushing boxes in a purely reactive fashion.

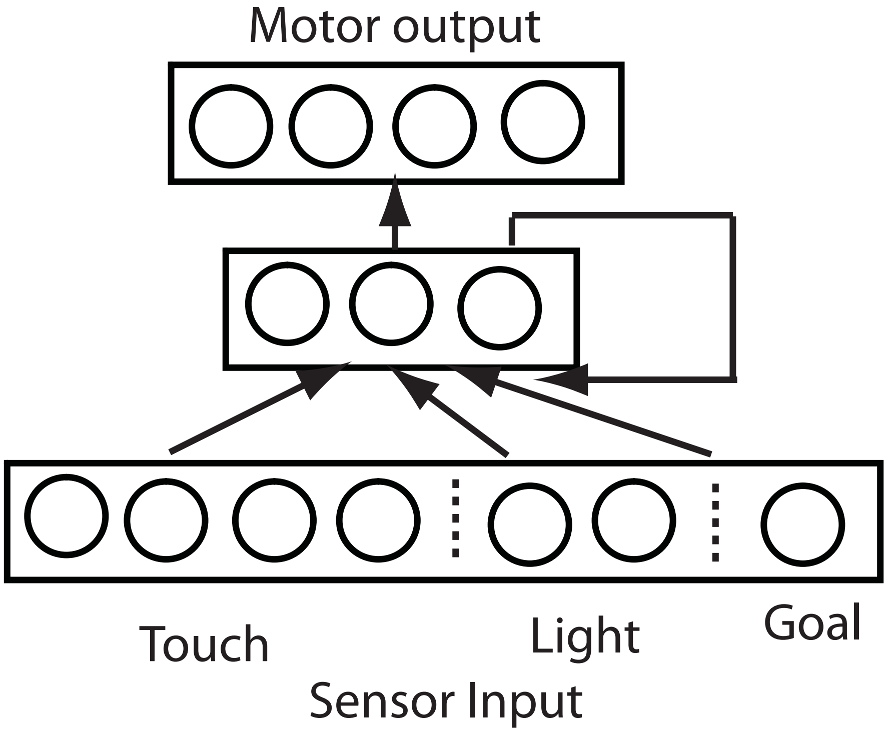


Figure 2.2: Recurrent Neural network. Adapted from [69]

The most conventional type of feedback is to reuse some of the controller networks activation values as extra inputs at a later (typically the next) point in time. A good example is the work by [69] who uses a recurrent ANN controller to guide a toy-car-like vehicle. The vehicles task is to keep moving around in an environment while minimizing contact with obstacles and periodically seeking/avoiding a light source placed in the environment. The controller network of fig. 2.2 controls the vehicles two motors and receives sensory input from a number of touch and light sensors as well as a special input determining its current goal. Other recurrent neural network based method include [101] [35] [61].

There are a number of modular ANN approaches to the control of autonomous agents [111] [112]. However the controlled agents are typically not able to adapt their behavior themselves. Adaptive Mixture of Experts (AME) such as [53][43] [102] also used modular ANN of the form of many expert with one gating network.

The advantage of AME is that Switching between experts is carried out by the gating network and it can be learned as well. AME suffer from the problems of built-in modularization which leads to problems with adaptation of the behavioral organization, concerning how and when to add or remove expert module.

Evolutionary Algorithm

Evolutionary computation is based on the mechanics of natural selection and process of evolution. Chromosome encode the potential solutions of the problem to solve. During the search, chromosome are combined and muted in order to find the best solution (although it is not guaranteed to find the optimal solution).

Meeden [69], for example, used an evolutionary algorithm to find suitable weight settings for the recurrent controller network for the robotic vehicle.Comparisons to the results achieved with conventional ANN learning showed that evolutionary algorithms can find suitable solutions more reliably in cases

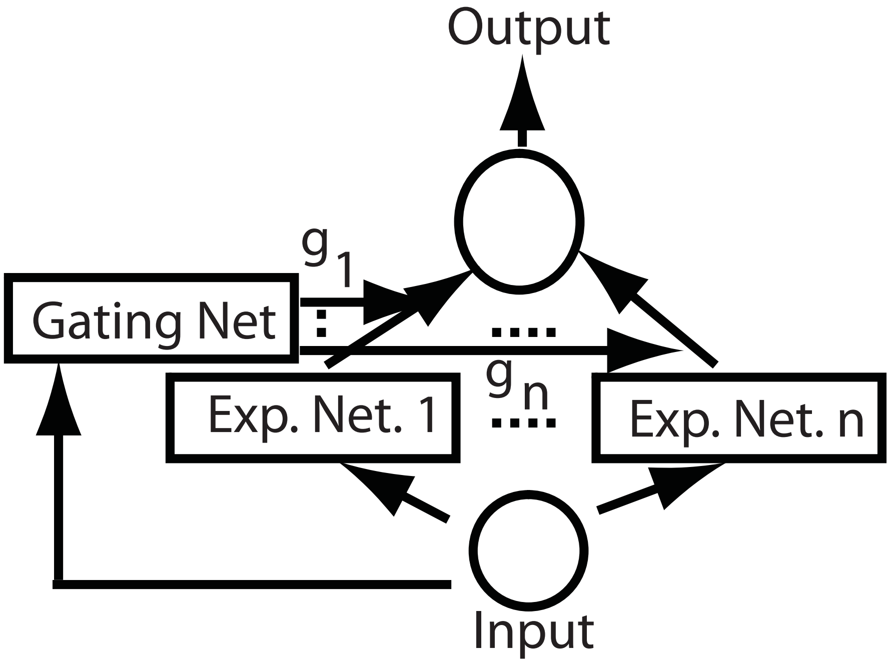


Figure 2.3: Adaptive mixer of experts. Adapted from [53]

where no sufficient reinforcement model is available. The problem with this type of learning in an individual robot, the evolution and evaluation of such a large number of controllers is often not possible/feasible due to real-time and memory restrictions.

2.5 Other approaches

Time window based adaptation methods include [11][30][38][58] [117][60]. Window based adaptation methods requires a preliminary investigation of the domain to determine the appropriate window size. Moreover, if the frequency of changes in the time series data are unpredictable, accuracy may deteriorate. In dual model based adaptation methods includes [13] [27]. In dual model based methods, separate models are used for short and long term learning. When short term model cannot infer with high confidence the system delegates the inference to a long term model where learning is done with the observations that were collected for a longer period of time. Long term module is not suitable for rapid adaptation as it perform with large number of observations. Adaptive control is also being tried for flight simulation or fault tolerant system[100][15]. These adaptive system deals how to detect sudden change for monitoring or how to control adaptively when there are faults in the control system.

Novelty detection is the identification of new or unknown data that machine learning system is not aware of during training. Novelty detection methods do so by modeling normal data and using a distance measure and a threshold for determining abnormality. Standard method include estimating data's probability density [66] [44], characterizing its geometry or identifying its support [23]. Novelty detection might be useful in behavior adaptation because novelty detection can be used to detects new behavior and then use them for adaptation. Here we discuss two statistical methods for novelty detection. Yamanishi [118] present

SmartSifter (SS), an outlier detection system based on unsupervised learning of the information source. Every time a datum is input, it is required to evaluate how large the datum has deviated compared to a normal pattern. The probability density over the domain of categorical variables is found using a histogram and a finite mixture model is employed for each histogram cell to represent the probability density over the domain of continues variables. Every time a datum is input, an on-line learning algorithm is employed to update the model. SS gives a score to each datum on the basis of the learned model indicating how much the model has changed after learning. A high score means that the datum is an outlier. The system was successfully tested on the network intrusion database,KDD Cup, 1999. In [110] Thomson presented a novelty detection method based on density estimation. In their work they compute the novelty threshold adaptively for any new dataset. They identify the appropriate threshold by computing the density estimate for each training example in the context of the new image. They used this adaptive thresholding technique for detecting novel rock and sediment features in rover space image.

2.6 Choosing Bayesian network

From the above discussion we get that it is difficult to adapt rapidly to follow user preference by using huge observation data for learning. We chose Bayesian network because it can represent degree of confidence for behavior decision as probability and can provide a confidence even with a small number of observations. Also Bayesian network is suitable for online interactive learning. A more details of benefits of using the Bayesian network representation are as follows:

- **Incorporation of prior knowledge.** Bayesian networks facilitate the translation of human knowledge into probabilistic form making it suitable for refinement by data.

- **Validation and insight.** In many cases a learned Bayesian network can be given a causal interpretation. Consequently a Bayesian network is more easily understood than black box representation such as neural networks. As an immediate by product the recommendations of a Bayesian network than those of a model justified only by its raw predictive performance is more logical. In addition users are more likely to gain insights from Bayesian networks.

- **Learning causal interactions.** Unlike purely probabilistic relationships causal relationships allow us to make predictions given direct interventions or manipulations of the world. Therefore, by learning with Bayesian networks there is a hope that we can make better predictions in the face of intervention. Learning causal relationships is crucial in scientific discovery where interventional studies are often expensive or impossible. Similarly, the ability to learn causal relationships is crucial for intelligent agents that must act in their environment on the basis of acquired knowledge.

- **Other benefits of using Bayesian networks for learning are derived from their probabilistic semantics.** Because sophisticated yet efficient methods have been developed for using a Bayesian network to answer probabilistic queries, they can be used both for predictive inference and diagnostic or abductive inference. This is in contrast to standard regression and classification method e.g. feed forward neural networks and decision trees that encode only the probability distribution of a target variable given several input variables. Whereas the Bayesian network representation can describe the casual ordering in the domain there are no restrictions as to the directions of the queries. Thus there is no inherent notion of inputs and outputs of the network. This property also allows Bayesian networks to reason efficiently with missing values by computing the marginal probability of the query given the observed values. One other cited benefit of the Bayesian network representation which derives from its probabilistic nature is that it can be used to determine optimal decisions.

2.7 Research focus

Here we give our a table showing goodness of different machine learning techniques. The items in the red shows that we focus our research in the area of Bayesian network and proposed significance evaluation of data for improved performance for rapid adaptation.

Table 2.1: Adaptation goodness of with different machine learning approach. Here ◯ represent good,△ represent fair and × represent need improvement.

Topic	RL	RNN	MNN	windowing	BN	Proposed
Reward Design	×	◯	◯	◯	◯	◯
Accuracy	◯	◯	◯	△	△	△
Robustness	△	×	×	×	◯	◯
Rapidness	×	×	△	△	△	◯
Obs. selection	×	×	×	◯	△	◯

2.8 Research map

Figure 2.4 provide a research map which indicates related research by reference number in accordance with our work. This research map is to visualize where our work might be placed. [108] reference shows our work. It is observed that methods which are fast to adapt donot deal well with interaction probably because of algorithmic limitation (bottom right). The methods which use traditional reinforcement learning and neural network donot do well both with rapidness and interaction(bottom left). Recent reinforcement learning with 'trace' or 'advice' providing interaction with the teacher and Bayesian network based interactive methods do well both in interaction and rapidness (top right). Top left shows the method which do well in interaction but not in in rapidness. We aimed at developing an algorithm that would both be rapid and interactive for Human-centered robot. Our current work shows that we could get good rapidness and we are working to integrate interaction with it. As we had chosen Bayesian network as our framework we aimed at producing a rapid behavior adaptation algorithm and we hope to bring good results.

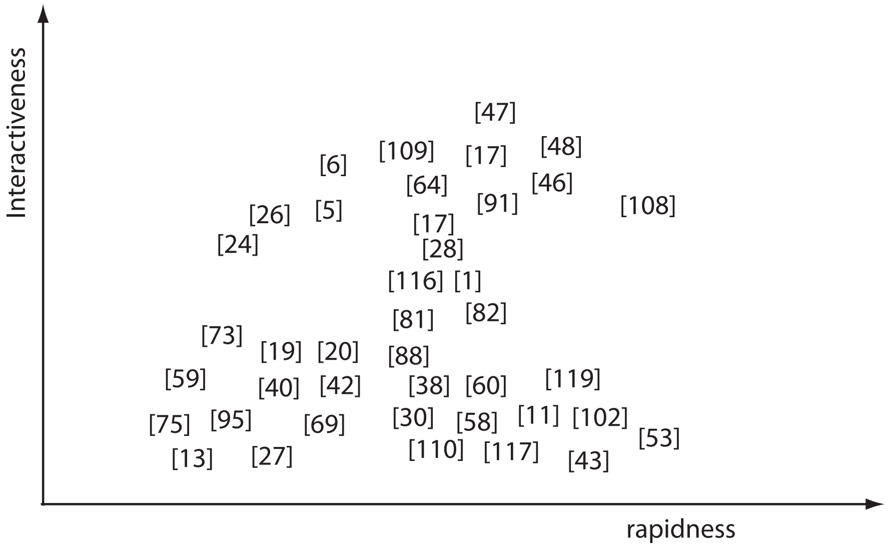


Figure 2.4: Shows the research map of the related reference in accordance with our work.

2.9 Conclusion

In this chapter we discussed about the previous work on policy learning. We also discussed about the shortcoming of those policy learning methods. And finally we discussed why we have chosen Bayesian network for rapid policy adaptation.

Chapter 3

Rapid Adaptation to user preference

3.1 Introduction

To make robots useful to non-technical users, learning from observation is an intuitive approach because robots do not require embedding all behaviors for all users. Robots learn novel behavior strategy using conventional methods that extract meaningful relation between observed sensor and user commands. When the user behavior preference changes frequently the robot has to adapt to new user behavior preference rapidly. In this thesis, as mentioned earlier, users preference indicates variation of behavior decision by the user even though identical sensor is observed. For example let us consider operation of a mobile robot; obstacles are approaching. An operator may prefer to avoid obstacles sometimes by turning left; sometimes by turning right. Robot's ability to rapidly adapt to user behavior preference is an important aspect of learning from observation because otherwise the user may be tired. However it is difficult to adapt to users behavior preferences rapidly with conventional methods. This chapter presents a rapid adaptation method of behavior preference based on Bayesian significance evaluation of experience data. Rapid adaptation to user preference cannot be achieved when data from every process cycle is used for learning because significant data are not differentiated with insignificant data. We propose a method to solve this problem by selecting significant data for the learning based on change in degree of confidence of the behavior decision. A small change in the degree of confidence can be regarded as reflecting insignificant data for learning, so that data can be discarded. Accordingly the system can avoid having to store too frequent experience data and the robot can adapt rapidly to changes in the users preference.Here, as already mentioned, rapidness is defined as the ratio of expected time to change user's preference by the user to the required time to adapt to preference by the robot.

The following section describe the problem with an example scenario. Section 3.3 describe the concept of significance evaluation. Section 3.4 describe the method, Section 3.5 describes the experiment to test its effectiveness, and section 3.6 presents experimental results and Section 3.8 concludes the paper with a summary and concluding remarks.

3.2 Definition of Preference, the problem by an example

To start with let us define user preference formally. We already know users preference indicates variation of behavior decision by the user even though identical sensor is observed.

A user demonstration data $\{d_i, b_j\} \epsilon D$ is represented by pair of sensor observation d_i and user behavior b_j where $d_i \epsilon S$ and $b_j \epsilon B$. Then the user preference can be defined by $P(B = b_j)$ such that $\sum_j P(B = b_j) = 1$. We explain the matter with the following illustration.

Let us consider that the B have two values b_1 and b_2 and b_1 indicates 'turn right' and b_2 indicate 'turn left'. Now at the the junction if the user take a 'turn right' we say that the user preference of turning right is $P(B = b_1)$. If for several trials the user prefer to turn right then user preference increase to turn right. If the user prefer to turn left then the preference of turning left is given by $P(B = b_2)$. At any time the user preference of turning right and turning left is

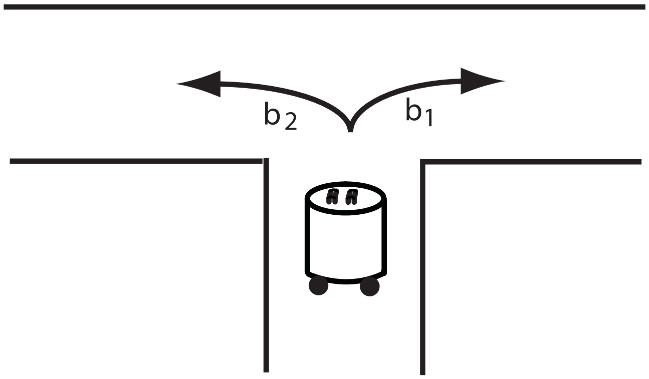


Figure 3.1: Illustration for understanding preference

given by $P(B = b_2)$ and $P(B = b_2)$.

As an example, imagine that we have a car robot and its driver changes frequently (e.g., it is a rental car). Also imagine that the teaching and learning are done through a driver assist system. The driver assist system learns from each drivers actions in order to act in accordance with his preference. For example, when a driver typically drives at a moderate speed, the driver assist system should recognize the drivers preference and give suggestive feedback to lower the cars speed when it becomes high. In contrast, under normal circumstances, the system should not suggest lowering the cars speed on a freeway if the driver typically drives close to the speed limit. If the drivers change frequently, the assist system has to adapt preference rapidly to every change.

From a brief introduction of Bayesian network we start describing problem. A Bayesian network is a directed acyclic graph consists of parent nodes representing causes and child nodes representing effects as shown in Fig.3.2. Each node can represent a multi valued variable, comprising of collection of a mutually exclusive propositions. Let the variable be labeled by capital letters (X, Y, Z_1, Z_2, Z_3) and their possible values by the corresponding lowercase letters (x, y, z_1, z_2, z_3). Each directed link $X \rightarrow Y$ is quantified by a fixed conditional probability table (CPT)

in which the (x,y) entry is given by

$$CPT_{(Y|X)} \equiv P(Y=y|X=x) = \begin{pmatrix} P(y_1|x_1) & P(y_2|x_1) & \dots & P(y_m|x_1) \\ P(y_1|x_2) & P(y_2|x_2) & \dots & P(y_m|x_2) \\ \vdots & \vdots & \ddots & \vdots \\ P(y_1|x_l) & P(y_2|x_l) & \dots & P(y_m|x_l) \end{pmatrix} \tag{3.1}$$

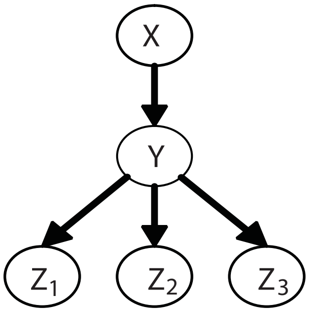


Figure 3.2: A Bayesian network

The CPT is calculated for all parent-child nodes. The reasoning can be expressed as

$$Bel\left(Y\right) = \beta\lambda\left(Y\right)\pi\left(Y\right), \tag{3.2}$$

where $\lambda\left(Y\right)$ represents the current strength of diagnostic support contributed by the children of Y given by $\prod_i \lambda_i\left(Y\right)$, $\pi\left(Y\right)$ represent the current strength of the causal support contributed by the parents of Y and β is the coefficient for normalization [83]. Elements of $Bel\left(Y\right)$ indicate the plausibility for each proposition of a node. When Y does not have any parent, $\pi\left(Y\right)$ is the prior probability of Y. The likelihood vector $\lambda_i(Y)$ is calculated as

$$\lambda_i(Y) = CPT_{(Z_i|Y)}\lambda(z_i), \tag{3.3}$$

where $CPT_{(Z_i|Y)}$ quantifies $Y \rightarrow Z_i$ link and $\lambda(z_i)$ is the observed input of Z_i. One of the advantages of Bayesian networks is that a robot can evaluate the vagueness of a behavior decision, and this leads it to ask questions and give suggestions to users [46]. For example, the robot should ask the user to confirm the behavior decision when the elements of $Bel(\mathrm{Y})$ are almost equal.

We emphasis the need of the change in degree of confidence based significance evaluation of data with an example as shown by Fig.3.3. According to definition the user's preference p_j is the probability to select behavior b_j. The robot observes the user's behavior b_j and gathers the sensor information d_i at the same moment. Let N be the number of observed data, N_j be number of observations of behavior b_j, and n_{ij} be the number of observation of behavior b_j for d_i. One of the simplest calculations based on the observation is

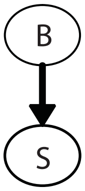


Figure 3.3: A simple Bayesian network

$$P\left(d_i|b_j\right) = P\left(S = d_i|B = b_j\right) = \frac{n_{ij}}{N_j}, \tag{3.4}$$

$$P\left(b_j\right) = P\left(B = b_j\right) = \frac{N_j}{\sum_j N_j}, \tag{3.5}$$

A problem arises with this simple calculation when the data is continuously input during the observation. Suppose that the user behavior changes between two behaviors, b_1 and b_2. When a rare but important behavior b_2 is observed much smaller time than b_1, that is $N_2 \ll (N_1 + N_2)$, the probability $P\left(B = b_2\right)$ is close to 0 and $P\left(B = b_1\right)$ is close to 1. This simple prior probability calculation by Eq(3.5) based on frequency of the number of data causes the problem because even if the conditional probability by Eq(3.4) shows a feasible value, the degree of confidence, $Bel(B)$, by Eq(3.2) becomes heavily biased. Consequently the system tends to output the most frequent command even though sensor input for rare behavior is given. This factor also causes another problem that the robot cannot adapt rapidly to changeable preference of the user.

In this chapter we are considering rapid adaptation to user's preference by a mobile robot. In the experiment the user teaches the mobile robot of his behavior in a three way junction(T-type). The user teaches the robot 'turn right' behavior for many occasions when coming to the junction from the same direction. And after that user changes his preference and started to teach 'turn left' behavior. Therefore the robot needed to adapt rapidly to the user user's preference. We propose a method in which the important observation is selected on basis of the change in the degree of confidence. We extend the experiment in a more complex scenario in which the user teaches the mobile robot of behavior 'go forward' for a long duration. And when the user changes behavior to 'turn left' the robot needed to adapt rapidly to the new user preference. We applied the change in the degree of confidence based significance evaluation in this case also. The next section discusses the concept of significance evaluation based on the change in the degree of confidence.

3.3 The concept of significance evaluation

To overcome the problem we proposed a method to evaluate significance of data based on the change in the degree of confidence. In this section we will describe the concept. Suppose that robot observe data continuously for learning where data comprised of user behavior and sensor information. Also say that the user has two behaviors, b_1 and b_2. The concept of the method is that when robot observes a datum it checks whether it is significant for learning. The significance evaluation is carried out by assigning user behavior to the parameters of a distribution, in this case to a binomial distribution as shown in the following Fig.3.4.

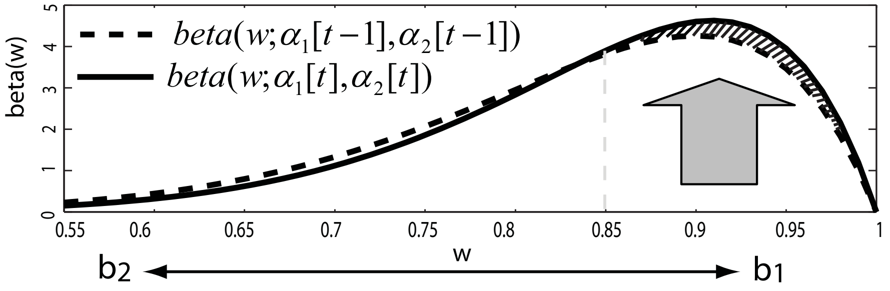


Figure 3.4: Illustrating concept of the change in degree of confidence

In the figure two end of w represents two user behaviors where peak of the distribution concentrate when one parameter is higher than other. Distribution parameters $\alpha_j[t]$ refers to a set of number of observations for the user behavior b_j at time t and these parameters represent the shape of the distribution. The distribution represents not only event probability among two behaviors, but also degree of confidence for the output probability. The system calculates the degree of confidence by integrating the area around peak of the distribution after each observation. The change in the two consecutive degrees of confidence, shown by the dashed area above arrow in the Fig.3.4, can be regarded as the importance of the observation to the learning process. When the change in the degree of confidence in two consecutive time steps is small, this situation is regarded as familiar; the experience data is considered insignificant for learning and be discarded. In contrast, when the robot detect a large change in the degree of confidence in two consecutive time steps, this situation is considered unfamiliar; the experience data is considered significant for learning and be accepted. This following section describe the detail method.

3.4 The significance evaluation method

The change in degree of confidence based significance evaluation is described in this section in detail. We adopted Beta and Dirichlet distributions to evaluate the significance of data. We explain our method with beta distribution for a decision follower that can change between two preferences. Later, we extend our system using the Dirichlet distribution for choosing among multiple preferences. We choose beta and Dirichlet distribution because with the help of them we can evaluate confidence even though the number of observation is small.

3.4.1 In the case of two preferences

We use a beta distribution to evaluate the significance of data based on changes in the degree of confidence when the behavior has two propositions. A beta distribution is a family of continuous probability distributions given by

$$f_b\left(\mathbf{w};\alpha_1,\alpha_2\right)=\frac{\Gamma(\alpha_1+\alpha_2)}{\Gamma(\alpha_1)\Gamma(\alpha_2)}w^{\alpha_1-1}(1-w)^{\alpha_2-1}=\frac{1}{Beta(\alpha_1,\alpha_2)}w^{\alpha_1-1}(1-w)^{\alpha_2-1} \tag{3.6}$$

where α_1 and α_2 are positive shape parameters, Γ is the gamma function, and *Beta* is the beta function. The beta distribution is uniform when $\alpha_1 = \alpha_2 = 1$. When $\alpha_1 = \alpha_2$, the probability distribution function is symmetric at about $w = 0.5$. If $\alpha_1 < \alpha_2$, the peak of the probability distribution function moves to the left; when $\alpha_1 > \alpha_2$, the peak probability distribution function moves to the right. Fig. 3.5 shows the beta distributions for three sets of α_1 and α_2.

In the figure3.5 w represents the order of the distribution. The two extreme values of w can represent two behaviors, for example, $w = 0$ can represent 'turn

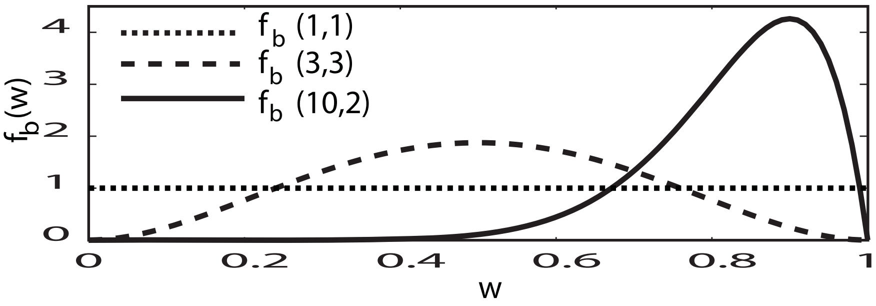


Figure 3.5: Beta density functions for different values of α_1 and α_2

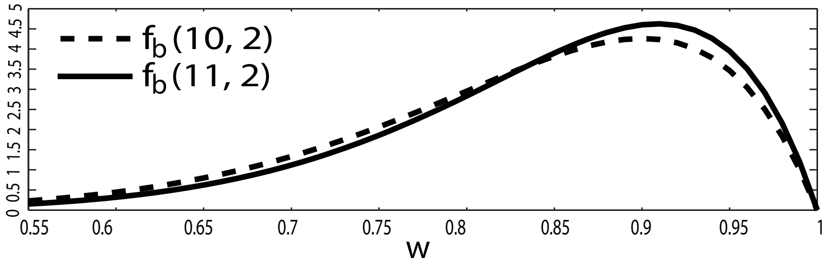


Figure 3.6: Two beta density functions with peak moved to the right

left' and $w = 1$ can represent 'turn right'. The area around the peak of the distribution represent confidences. The distribution parameter α_1 and α_2 are expressed as $\alpha_1 = 1 + N_1$ and $\alpha_2 = 1 + N_2$. The system increases α_1 or α_2 according to the observed behavior. The degree of confidence is calculated by integral of the distribution as

$$C_1[t] = \int_{1-u}^{1} f_b\left(w; \alpha_1[t], \alpha_2[t]\right) dw, \tag{3.7}$$

$$C_2[t] = \int_{0}^{u} f_b\left(w; \alpha_1[t], \alpha_2[t]\right) dw, \tag{3.8}$$

where$f_b\left(w; \alpha_1[t], \alpha_2[t]\right)$ is the beta distribution, $\alpha_j[t]$ is α_j at time t, $0 < u \leq 0.5$, integration is done from $1 - u$ to 1 when the peak moves to the right and 0 to u when the peak moves to the left, $C_1[t]$ represents the confidence when the peak moves to the right at t and $C_2[t]$ represents confidence when the peak moves to the left at t. Suppose that the system accepts data for learning b_1, and it increases α_1. The system calculates the degree of confidence at $t - 1$ and t by Eq(3.7). We think the difference between the two degrees of confidence can be regarded as the importance of the observation for learning process. To evaluate the significance of the observation data, the following criteria are calculated.

$$E_j = |C_j[t] - C_j[t-1]|, \tag{3.9}$$

Here, E_j represents the change in degree of confidence for evaluating the significance of the data. Let $\mathbf{v}[t] \epsilon \{d_1, d_2, \ldots, d_n\}$ be the observation of the sensor at time t. Let $o[t] \epsilon \{b_1, b_2, \ldots, b_m\}$ be the observation of the user's behavior at time t. Then, we can define data as $D[t] = \{\mathbf{v}[t], o[t]\}$. If E_j is less than a specific threshold, the system evaluates $D[t]$ as insignificant for learning and discard. Let θ be the threshold. Data $D[t]$ are significant for learning and accepted when

$E_j \geq \theta$, and are insignificant for learning and discarded when $E_j < \theta$. The steps for learning b_j are as follows

(1) Receive $D[t]$ and assign α_j as the number of observation of b_j

(2) Calculate degree of confidence $C_j[t]$

(3) Calculate change in the degree of confidence, E_j

(4) If $E_j \geq \theta$, then $D[t]$ is significant; $N_j^{sig} := N_j^{sig} + 1$ and go to Step 1

(5) Else $D[t]$ is insignificant; discards it and go to Step 1

The θ can vary from application to application. For example, an application with a very high input frequency will have a different threshold (low value) from one with a very low input frequency (relatively high). For rapid adaptation, the area and threshold should be determined by experimentation, as discussed in section 3.5. Here, we explain the probability calculation based on the concepts explained above. Let N_j^{sig} be the number of significant observations while the user behavior b_j is observed. Then we get

$$P\left(b_j\right) = P\left(B = b_j\right) = \frac{N_j^{sig}}{\sum_j N_j^{sig}} \tag{3.10}$$

Now let us explain the consequences of the significance evaluation method. We consider that user changes between two behaviors b_1 and b_2; the frequency of observation of b_2 is less than that of b_1. Even though the importance of two behavior is same, the expected likelihood of selecting b_2 would be less than that of b_1. However N_2^{sig} can be almost equal to N_1^{sig} because significance evaluation method discards most of insignificant observation data for b_1. Then prior information becomes uninformative because prior probabilities for both behavior become almost equal. Therefore, the degree of confidence remains unbiased for both behaviors, and robot can adjust rapidly to the user's new preference with a few observations.

3.4.2 In the case of multiple preferences

The previous section explained the case of a behavior node with two preferences. In this section, we extend the algorithm to handle multiple preferences. We use a Dirichlet distribution to evaluate the significance of data based on changes in the degree of confidence instead of Eq(3.9). A m-directional Dirichlet distribution for $\mathbf{w} = \{w_1, w_2, \ldots, w_m\}$, is given by

$$f_d\left(\mathbf{w}; \alpha_1, \ldots, \alpha_m\right) = \frac{1}{Z} \prod_k w_k^{\alpha_{k-1}}, \tag{3.11}$$

where,

$$Z = \frac{\prod_{k-1}^{m} \Gamma\left(\alpha_k\right)}{\Gamma\left(\sum_{k-1}^{m} \alpha_k\right)}, \tag{3.12}$$

is a normalization factor, Γ is the gamma function and the $\mathbf{m}$ parameters α_m are assumed to be positive.

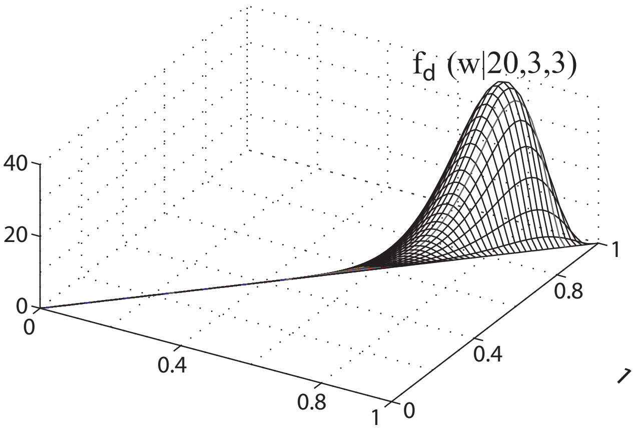


Figure 3.7: Dirichlet density functions with peak moved with corresponding w

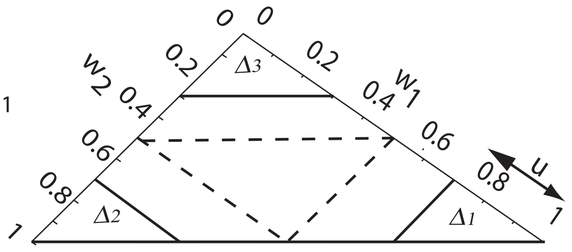


Figure 3.8: Three areas $(\Delta_1, \Delta_2, \Delta_3)$ where the probability density can be integrated.

When α_1 becomes larger than the other Dirichlet parameters, the peak of the distribution moves within a small area at the end of corresponding variable as shown by Fig.3.7. The system calculates the degree of confidence at $t-1$ and t. Confidence at time t is calculated as

$$C_j[t] = \int_{\Delta_j} f_d\left(\mathbf{w}; \alpha[t]\right) d\mathbf{w}, \tag{3.13}$$

where $f_d\left(\mathbf{w}; \alpha[t]\right)$ is the Dirichlet distribution at time t and Δ_j represents area of integration where the peak of the distribution is moved by observing b_j as shown by Fig.3.8 where u is the integration limit over w. The change in the two degrees of confidence can be regarded as the importance of the observation to the learning process. To evaluate the significance of the observation data, the criteria

$$E_j = |C_j[t] - C_j[t-1]|, \tag{3.14}$$

is calculated. As in the previous case C_j and E_j are calculated when the behavior b_j is observed. When $E_j \geq \theta$, data $D[t]$ are significant for learning and accepted, that is, $N_j^{sig} := N_j^{sig} + 1$, and data are insignificant for learning and discarded when $E_j < \theta$.

3.5 Experimental setup

3.5.1 Determining evaluation parameter

An multinomial behavior node, which used a Dirichlet distribution for evaluation, is used for experiment. Before starting the experiment, we needed to fix the areas of integration $(\Delta_1, \Delta_2, \Delta_3)$ and threshold θ for the significance evaluation. We tested this method for different non-overlapping areas and different thresholds. The selection of area of integration and threshold is explained with help of Fig. 3.9, Fig. 3.10 and Fig. 3.11. As shown in Fig. 3.9, when the area of integration increased, the system needed fewer data to reach significance evaluation criteria for fixed threshold. Therefore, it was reasonable to use a larger area of integration. As shown in Fig. 3.10, when the area was fixed and the threshold was increased, the time required to reach the significance evaluation criteria is decreased. Therefore, for rapid adaption, a higher threshold should be used. However, when we used a higher threshold value, the system failed to reach a satisfactory level (0.95) of probability of behavior as shown in Fig. 3.11. The area of integration and threshold determine the required rapidness of the system and thus can be adjusted according to application. For our current experiments, therefore, we set the area of integration limit u from 0.0 to 0.5 and the threshold to 1×10^{-6}.

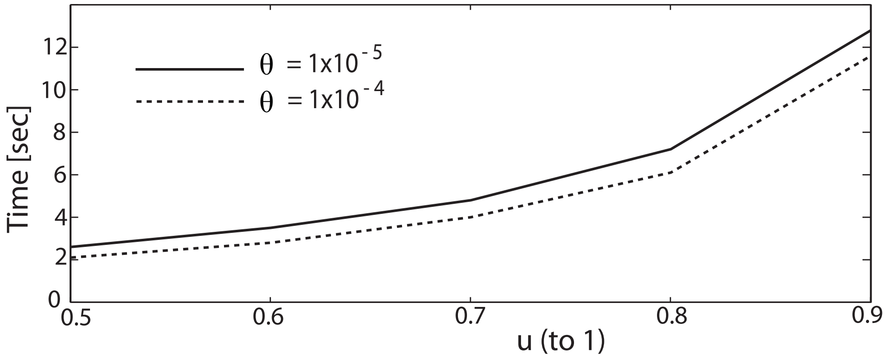


Figure 3.9: Relationship between area of integration and learning time for fixed θ

3.5.2 Experimental setup

We developed a teaching and learning system that incorporated the significance evaluation method based on the change in the degree of confidence. The user controls the teaching information by using a joystick that corresponds to the behavior node. We trained the system for three simple behaviors: go forward (b_1), turn left (b_2), and turn right (b_3). In the experiment, we used a Bayesian network consisting of eight distance sensor nodes and a behavior node as shown in the Fig. 3.12. The input data, $D[t] = \{\mathbf{v}[t], o[t]\}$, was given to the robot at all the time. The distance sensors were used to measure the distance to obstacles. The user operated the robot with joystick. Joystick inputs were translated into

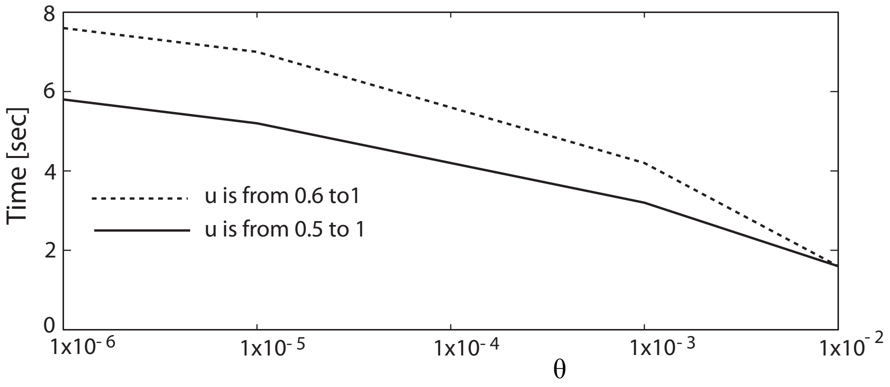


Figure 3.10: Relationship between θ and learning time for fixed Δ (given by u)

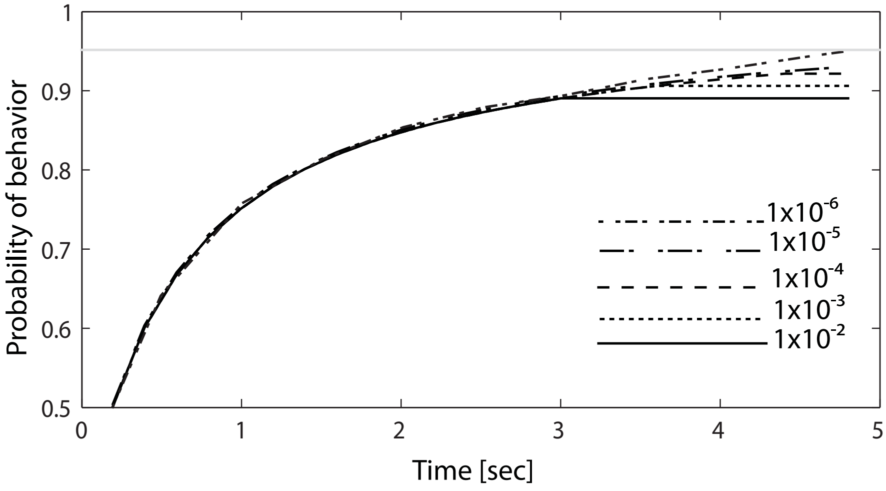


Figure 3.11: Threshold values and corresponding probability of behavior over time

discrete instructions by using a predetermined threshold.

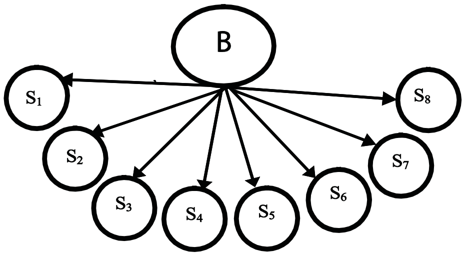


Figure 3.12: A Bayesian network used in the experiment

3.5.3 Experimental environment for virtual mobile robot

To carry out experiments on teaching and learning, we developed a virtual environment for a mobile robot and a user interface.The experimental environments as shown in Fig. 3.13 and 3.14 was prepared using Webot [70] real-time simulation software. The 'pioneer' robot model we used had eight front sensors. The pioneer robot model has length 47 [cm], width 38 [cm], height 24.5 [cm] and clearance 6.5 [cm]. The robots wheel diameter is 191 [mm] and width of the wheel is 50 [mm]. The robot's eight front sonar distance sensors mounted on the front from left to right were used for the experiments. The first virtual experimental environment had an enclosed area of 8 [m]× 8 [m]. Width of each lane was 2.5 [m]. Junction's tail path was 5[m].

The second experimental environment had an enclosed area of 8 [m]× 8 [m]. The length and width of the initial forward path were 4.8 [m] and 1.5 [m]. The length and width of the next path (left to initial position) were 3.75 [m] and 1.8 [m], and the length and width of the third path (left and then right to initial position) were 3 [m] and 1.5 [m]. For the second experiment when data for a new behavior is evaluated as significant and is added to the secondary database, one datum $D[t]$ with oldest t is deleted, if existed. This ensures that the maximum number of significant data to adapt any user behavior remains same. One of the related method is widowing technique [60]. But our method has the advantage that it can select significant data by using evaluation criteria given by Eq(3.9) or Eq(3.14). Evaluation criteria not only decide about the size of the learning database(window) but also decide when data become insignificant and should be discarded.

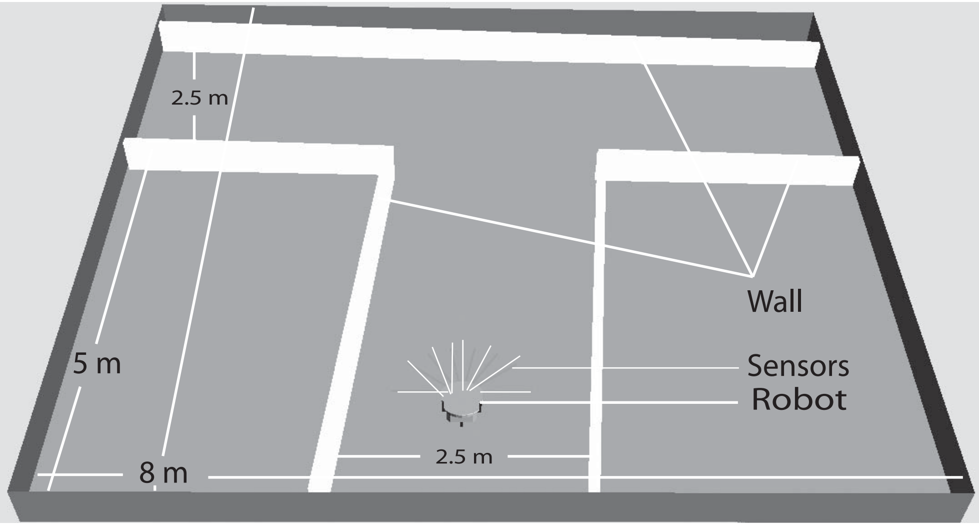


Figure 3.13: First experimental environment with virtual mobile robot

3.5.4 Experimental environment for real mobile robot

A pioneer-based performance PeopleBot robot was used as shown in Fig. 3.15. The PeopleBot robot has length 47 [cm], width 38 [cm], height 124 [cm] and clearance 3.5 [cm]. The PeopleBot robot was commercially available for the last few years and has been extensively used in robotics research (e.g. [8][2][36]). The robots wheel diameter is 191 [mm] and width of the wheel is 50 [mm]. The robot's eight front sonar distance sensors mounted on the front from left to right were used for the experiments. A joystick was used to control the robot remotely.

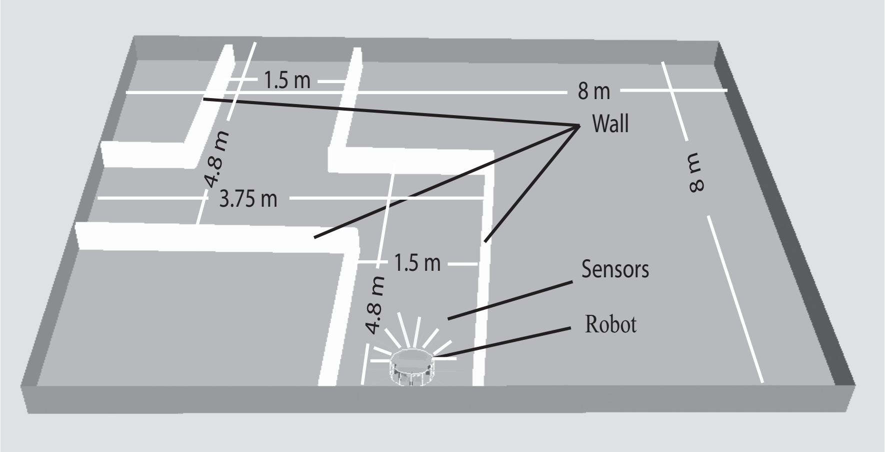


Figure 3.14: Second experimental environment with virtual mobile robot

After the distance sensor values were collected, each value was translated into a state value. Two state values were used for each distance sensor $\mathbf{v}[t] = \{d_1, d_2\}$. When the distance was less than 1 [m], d_1 was used; otherwise, d_2 was used. This threshold was the same for all sensors. The translational velocity of PeopleBot was 90 [mm/sec], and its rotational velocity was 15 [deg/sec]. The frequency of observation was 5 [Hz]. The experiment used a corridor environment. The length of the first part of the corridor was 8.8 [m], length of the second part of the corridor was 5 [m], length of the third part of the corridor was 4 [m] and width of the corridor in all csaes was 2 [m]. The significant data storage method was the same as the second virtual experiment.

3.6 Experimental results and discussion

3.6.1 Experimental results in the case of virtual mobile robot

In the first virtual experiment we considered rapid adaptation to user's preference by a mobile robot. In the experiment the user taught the mobile robot of his behavior in a three way junction. The user taught the robot 'turn right' behavior (b_3) for many occasions when fetching the junction. And after that user changes his preference and started to teach 'turn left' behavior (b_2). The experimental result is shown in Fig.3.16. The vertical axis represent N_j^{sig} and horizontal axis represent behavior execution time. Each time the user performed behavior b_3 the system increased α_3. The system started with uniform α_j and used Dirichlet distribution for evaluation. Along with the behavior b_3 each time the system calculated E_3 for evaluation. Until E_3 became less than threshold the system increased α_3 and used observations for learning. When the user changed preference and started behavior b_2 the system could automatically evaluate new data using E_2. Evaluation method bounded the number of significant data used

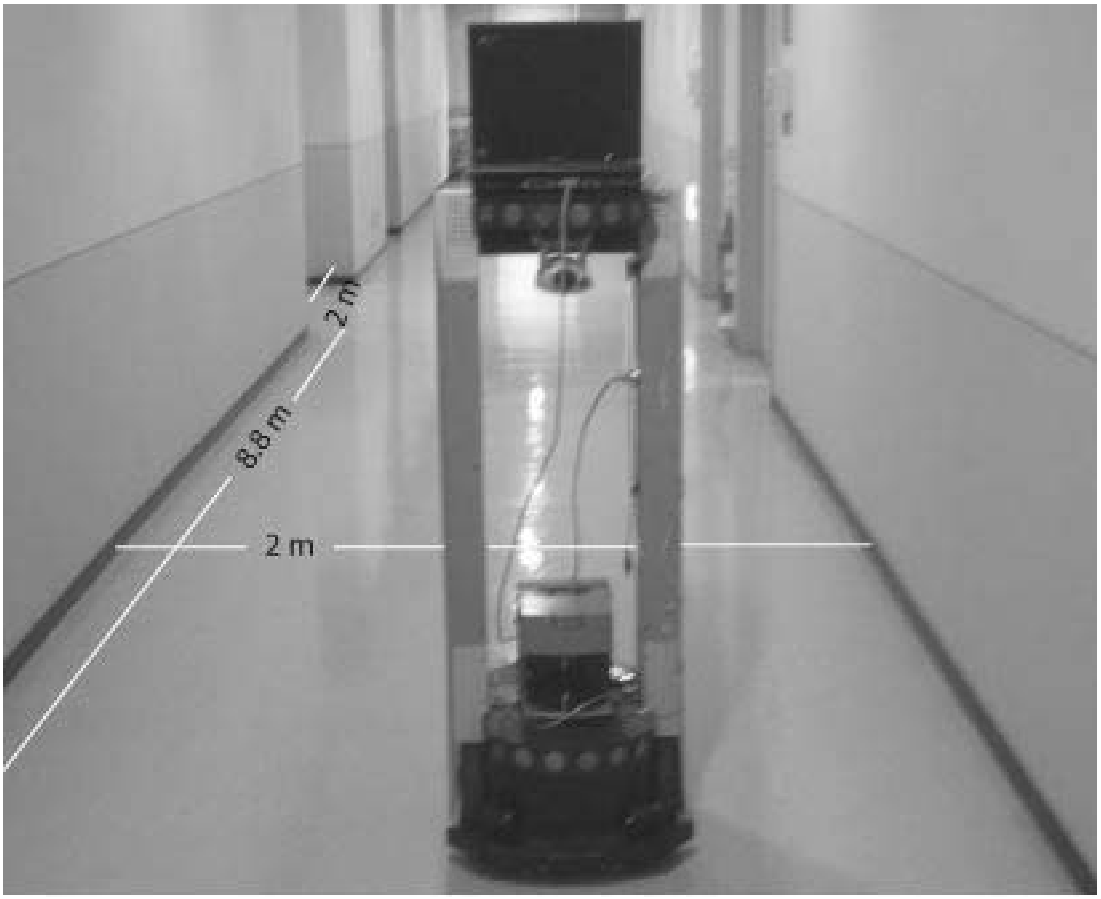


Figure 3.15: A PeopleBot in the real experimental environment

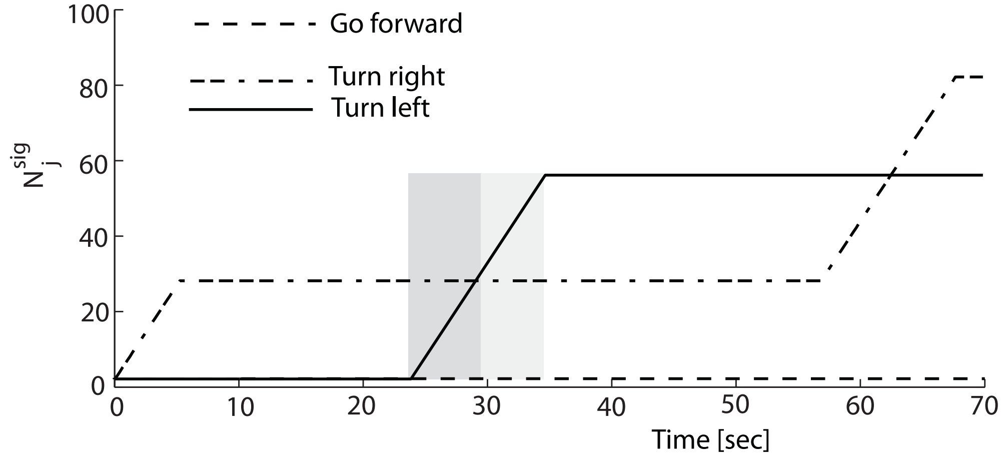


Figure 3.16: Significant data for preference adaptation

for learning and thereby shortening the time for adaptation. The shaded region in the Fig.3.16 shows that preference adaptation was achieved in 6.5 [sec].

In the second virtual experiment the user taught preference while the robot was running in the environment. Fig.3.17 shows the results of significance evaluation of data using the virtual mobile robot. The horizontal axis represents the time of observation. The vertical axis represents the number of experience data used for learning. The dashed line represents the number of experience data used for learning b_1 (go forward). When the change in the degree of confidence was below the threshold, the system evaluated data as insignificant (the flat portion of the line). The significance evaluation continued as long as the user did not change his preference. When the user changed his behavior to ”turn left”, the change in the degree of confidence was large and the system subsequently started to accept experience data for b_2. The system quickly adapted to the user’s new preference by learning with behavior b_2. The sloping lines show these changes. When learning b_3 (turn right), the change in the degree of confidence was large and data were accepted for b_3. b_3 was learned until the change went below threshold again, after which the data for this behavior was discarded.

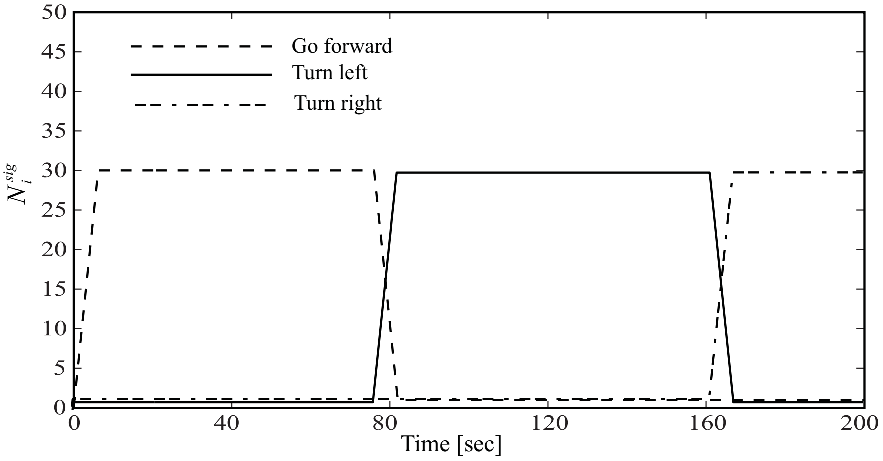


Figure 3.17: Uses of data for learning in a virtual environment

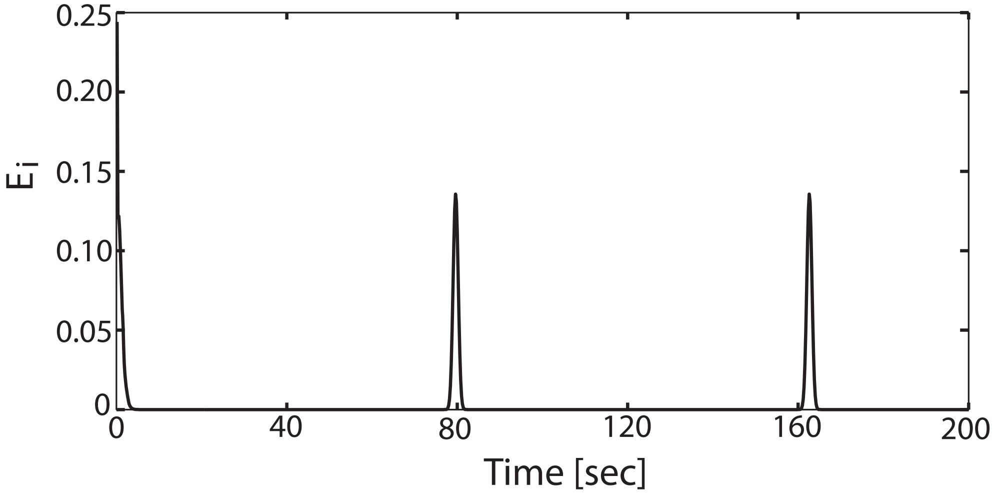


Figure 3.18: Change in the degree of confidence while adapting to new preference

Fig.3.18 shows the change in the degree of confidence (E_j) while adapting to new preferences. The figure shows that data for new behavior is evaluated as significant because the change in the degree of confidence is increased over threshold. The E_j indicates the highest value when the number of significant data for new behavior (α_2) becomes larger than that of old behavior (α_1). Afterwards as α_2 becomes larger than α_1, increasing α_2 more make the behavior more familiar and E_j is decreased. When E_j goes below the threshold, the data is evaluated as insignificant.

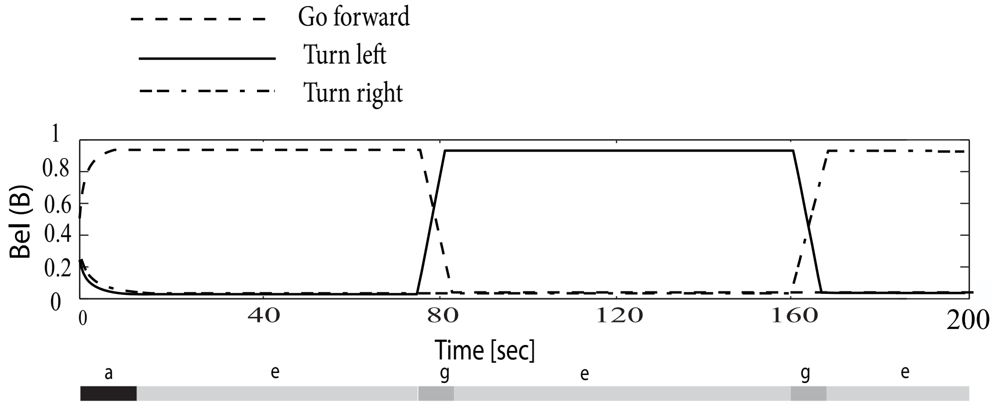


Figure 3.19: Progress of learning with significance evaluation by a virtual mobile robot

Fig. 3.19 shows how the probability of behavior changes over time in a virtual experiment. The horizontal axis represents the time of observation, and the vertical axis represents the probability of the learning behavior. Experience data was used and Bel(B=b_1) increased until the change in the degree of confidence fell below the threshold for behavior b_1 (region a in the bar). New data was not used when the change was below the threshold for behavior b_1, and Bel(B=b_1) remained the same (region e in the bar). When the user changed behavior from b_1 to b_2, the change in degree of confidence became large, data was accepted for the new behavior and Bel(B=b_2) increased and Bel(B=b_1) decreased as (region g in the bar). Our system quickly adapted to the user's new preference again when it changed behavior to b_3 by rapidly adjusting Bel(B=b_3). In this case the rapidness is 20 where expected time to change user's preference by the user is 75 seconds and expected time to adapt to the new preference by the system is 3.75 seconds.

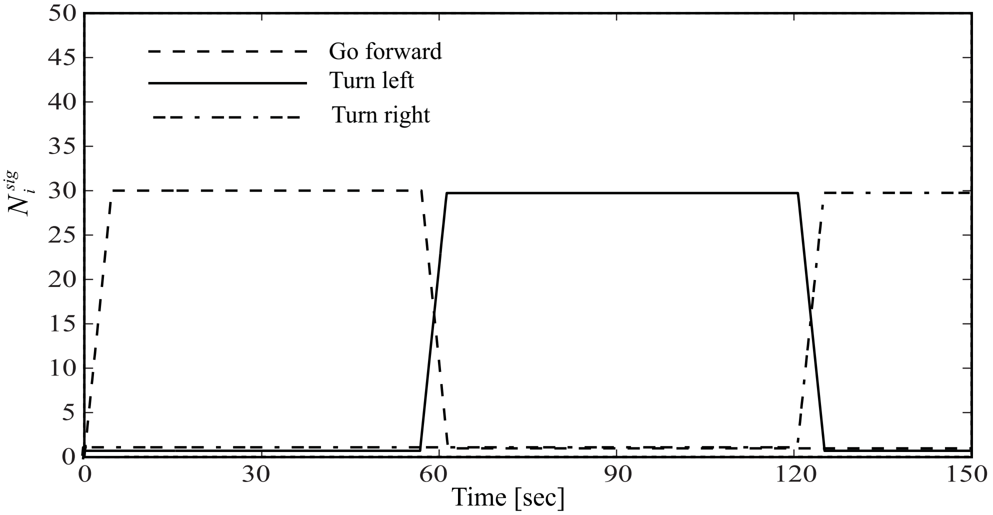


Figure 3.20: Uses of data for learning with the real mobile robot

3.6.2 Experimental results in the case of real mobile robot

Fig. 3.20 shows experience data evaluation result for the real mobile robot. The horizontal axis represents the time of observation. The vertical axis represents N_j^{sig}. The solid line represents the number of data points used for learning behavior b_1 (go forward). When the change in degree of confidence was below the threshold, the system evaluated data as insignificant (the flat portion of the line). When the user changed behavior to "turn left", the change in the degree of confidence was bigger than the threshold, and the system started to accept data for b_2. The system quickly adapted to the user's new preference by learning with behavior b_2. The sloping lines show these changes. When learning b_3 (turn right), the change was large and data were kept for b_3. Behavior b_3 was learned until the change fell below the threshold again, after which the data for this behavior was evaluated as insignificant.

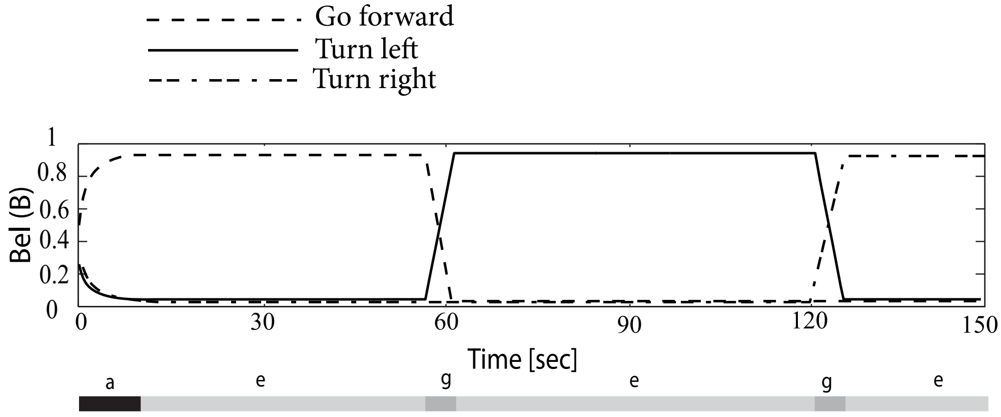


Figure 3.21: Progress of learning with significance evaluation with Peoplebot. The bar shows the regions of learning (a), insignificant (e), rapid adaptation to a new preference (g), insignificant (e), rapid adaptation to another new preference (g), and insignificant (e)

Fig.3.21 shows how the probability of user behavior changes over time. The horizontal axis represents the time of observation, and the vertical axis represents the probability of the learning behavior. Data was used and $Bel(\mathrm{B}=b_1)$ increased until the change in degree of confidence went below the threshold for behavior b_1 (region a in the bar). No more data was used after the change went below the threshold for behavior b_1, and $Bel(\mathrm{B}=b_1)$ remained the same (region e in the bar). When the user changed behavior from b_1 to b_2, the change in the degree of confidence became larger than the threshold and data was accepted for the new behavior. $Bel(\mathrm{B}=b_2)$ increased and $Bel(\mathrm{B}=b_1)$ decreased (region g in the bar). In this case the rapidness is 16.

Fig.3.22 shows the changing state more closely. This experiment examined the case in which a user behavior changed before the previous behavior became stable. In Fig. 3.22, the user preference initially was b_1 and when the system it

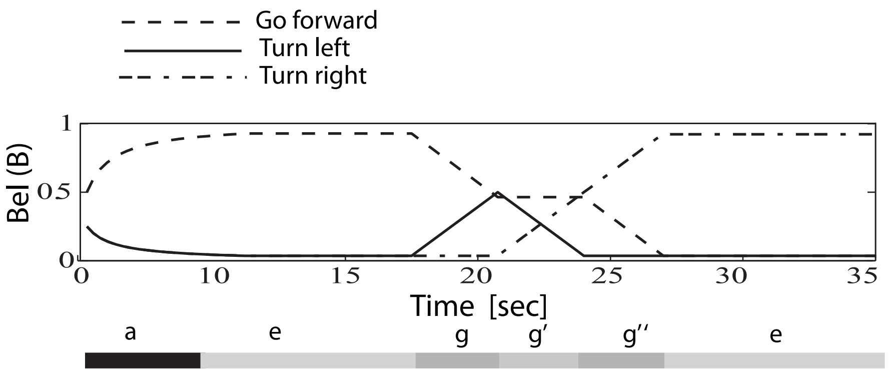


Figure 3.22: Progress of learning with significance evaluation in more detail. The bar shows the regions of learning (a), insignificant (e), rapid adaptation to a new preference (g, g',g''), and insignificant (e).

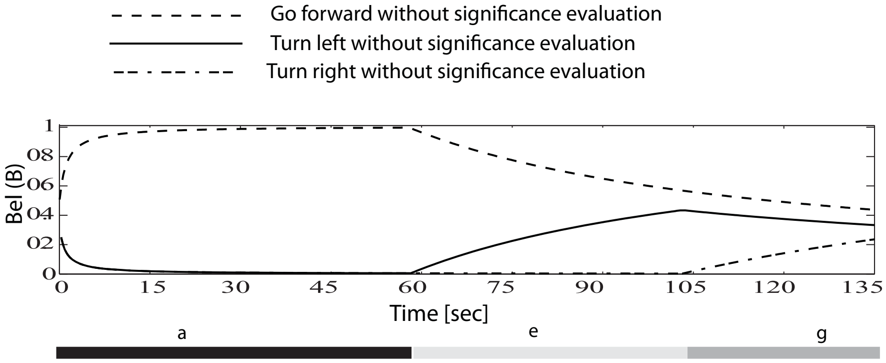


Figure 3.23: Progress of learning without significance evaluation. The bar shows the regions of learning (a) and slow adaptation to a new preference (e, g)

gained high confidence, the data was discarded for behavior b_1. When behavior changed to b_2, *Bel*(B=b_2) started to increase. However, the instruction was not enough and did not reach a high confidence level (for *Bel*(B=b_2)). Before the change in the degree of confidence went below the threshold for b_2, the user behavior changed to b_3. Hence, *Bel*(B=b_2) decreased and *Bel*(B=b_3) increased. The user continued with behavior b_3, and the change in the degree of confidence went below the threshold. Data was subsequently evaluated insignificant and discarded.

Fig. 3.23 shows the learning progression for *Bel*(B=b_1), *Bel*(B=b_2), and *Bel*(B=b_3) without the use of our method. The system gradually decreased *Bel*(B=b_1) when the behavior changed to b_2 (region e in the bar). It did not adapt to the change in user preference in 40 [sec]. When another change in behavior occurred, *Bel*(B=b_3) started to increase (region g in the bar), but the system failed to recognize the previous change as a user's preference. In the case of without significance evaluation the rapidness become 1 as per definition.

3.7 Evaluation of Rapidness

The rapidness of the adaptation depend on the θ and Δ. Here we provide a comparative figure with different θ and Δ. As we mentioned before according to the need of the application the value of θ and Δ can vary. This comparison is prepared when the observation frequency was 20 [Hz].

Figure 3.24 shows the rapidness of our method with different observation frequencies for two behaviors with one minute demonstration each when θ was $1 \times 10^{-6}$and for Δ area of integration limit u was from 0.0 to 0.5. In the figure δ represent ratio of rapidness of without to with significance evaluation method and ν represent frequency of observation. Fig. 3.24 shows that our method works better in high frequency. And as the frequency of observation decreases the rapidness also decreases and become equivalent to the method without significance

Table 3.1: Adaptation time with different θ and Δ_{0tou} with significance evaluation method

θ	Δ	adaptation time (sec)
0.01	0.5	0.50
0.001	0.5	1.50
0.0001	0.5	2.25
0.00001	0.5	3.50
0.000001	0.5	3.75
0.000001	0.4	5.25
0.000001	0.3	9.50
0.000001	0.2	20.90
0.000001	0.1	75.00

evaluation. This is because significance evaluation method requires a fixed number of observation with particular θ and Δ and if the demonstration time is not enough to get the required number of data then our become equivalent to without significance evaluation method.

3.8 Conclusion

We described an experience data management system for rapid adaptation to changes in user preferences for online teaching and learning. The degree of confidence is used for managing the experience data for learning. The system uses the change in the degree of confidence for evaluating the significance of the experience data. A small change in the degree of confidence implies that the data has little effect on the learning process. The system therefore discards data if the change in the degree of confidence is below a certain threshold and stores data if the change is above the threshold. This algorithm enables a robot to adapt rapidly to changes in the user's preference without having to store an enormous amount of data. Testing using a virtual and an actual robot incorporating this algorithm in an interactive teaching and learning environment showed that the time required to adapt to changes in user preferences among "go forward", "turn right", and

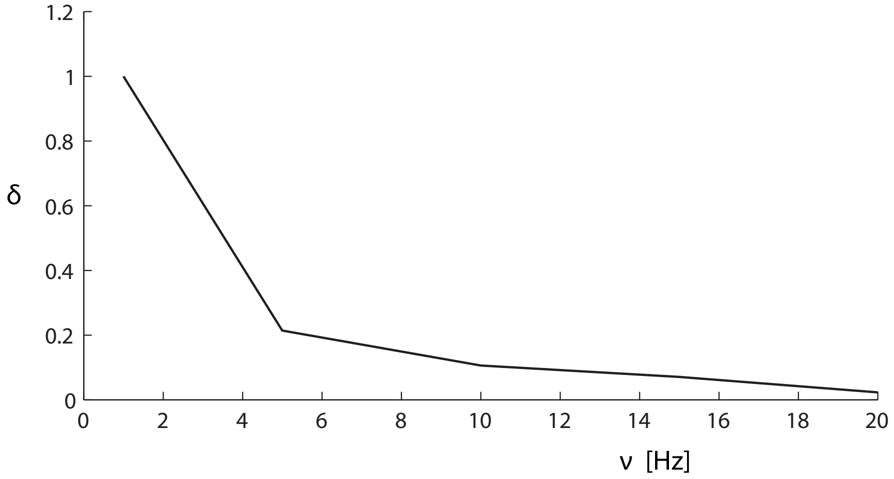


Figure 3.24: Rapidness with different observation frequency

”turn left” was 3.75 [sec] when the frequency of sensor observation was 5 [Hz], the translational velocity was 90 [mm/sec], and the rotational velocity was 15 [deg/sec]. When the frequency of observation was set to 20 [Hz], the actual robot could adapt within 1 [sec]. We think that the frequency of observation determines the time required to reach the threshold for evaluating data to be insignificant and hence the speed of adaptation. This should not be a problem because frequency of observation is almost higher than the frequency of changes of user preferences.

The work presented in this chapter evaluates data based on the behavior observation. As a result we could only handle prior probability and take the advantage of uninformative prior probability. In the following chapter we are considering significance evaluation for each sensor proposition. This will ensure that only significant sensor observation will be used for learning and we could take advantage of both prior and conditional probability which will ensure that our system is capable of handling rapid policy adaptation.

Chapter 4

Rapid adaptation to user policy

4.1 Introduction

This chapter presents a rapid adaptation method of behavior policy for mobile robots teleoperated by an operator. In our previous work [105] we proposed a method to manage experience data with evaluation of significance based on a concept of change in degree of confidence for behavior decision. Using that method, the robot adapted to a new preference by overriding the previous preference after evaluating the significance of its user-behavior observations. In that work, we only handled prior probability and hence only user preference could be adapted. We could not handle conditional probability and hence policy adaptation. That was a problem. In this chapter, we have solved that problem by using the significant evaluation method on sensor observation data.

4.2 The problem and an example

Policy is the mapping from world states to actions. Demonstrations provide the robot with a dataset consisting of state-action pairs representing examples of the desired behavior. The robots goal is to use this information to learn a policy, which enables the robot to select an action based upon its current world state.

Let us define policy formally. A user demonstration $\{d_i, b_j\}\epsilon D$ is represented by pair of sensor observation d_i and user behavior b_j where $d_i\epsilon S$ and $b_j\epsilon B$. Demonstration data D is used to directly approximate the underlying policy or mapping function from the robot's sensor observations to user behavior ($\pi : S \rightarrow B$) as $B = \pi(S)$.

As an example, in Robocup soccer when the robot get hold of the ball it start to 'approach' the goal of the opponent. When the robot finds opponent it has to change policy to 'avoid' the opponent. In the game the policy might change from approach to avoid and vice versa.

4.3 Significance evaluation for policy adaptation

The distribution parameter α_j is expressed as $\alpha_j = 1 + n_{ij}$. The system increases α_j according to the observed sensor proposition. When α_j becomes larger than the other Dirichlet parameters, the peak of the distribution moves within a small area at the end of the corresponding variable, as shown by the circle in Fig.4.1.

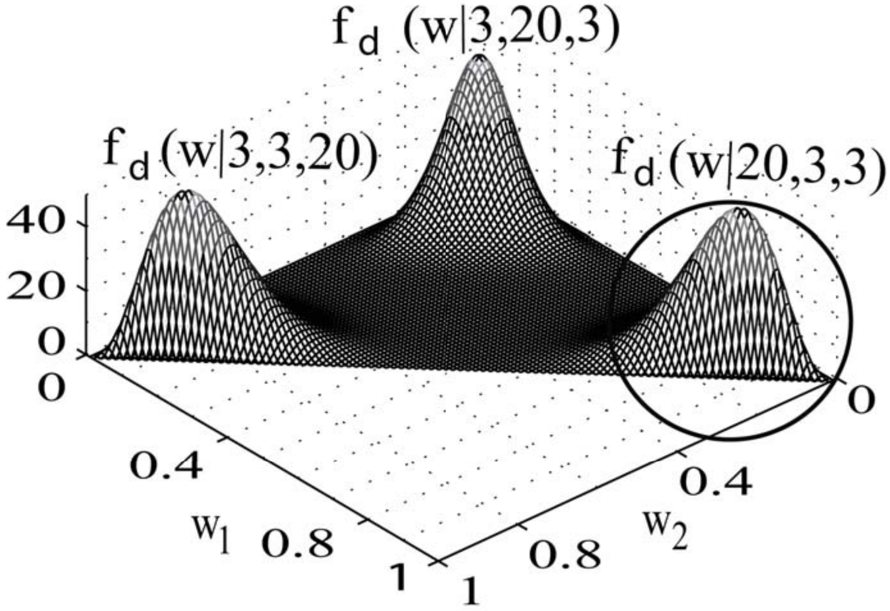


Figure 4.1: Three superimposed Dirichlet density functions with three parameters.

The change in the degree of confidence is calculated as per Eq.3.14 in chapter3. The C_j and E_j are calculated when the sensor proposition d_i is observed for b_j. When $E_j \geq \theta$, data $D[t]$ are significant for learning and accepted, and data are insignificant for learning and discarded when $E_j < \theta$. The steps for evaluating sensor observation d_i are as follows

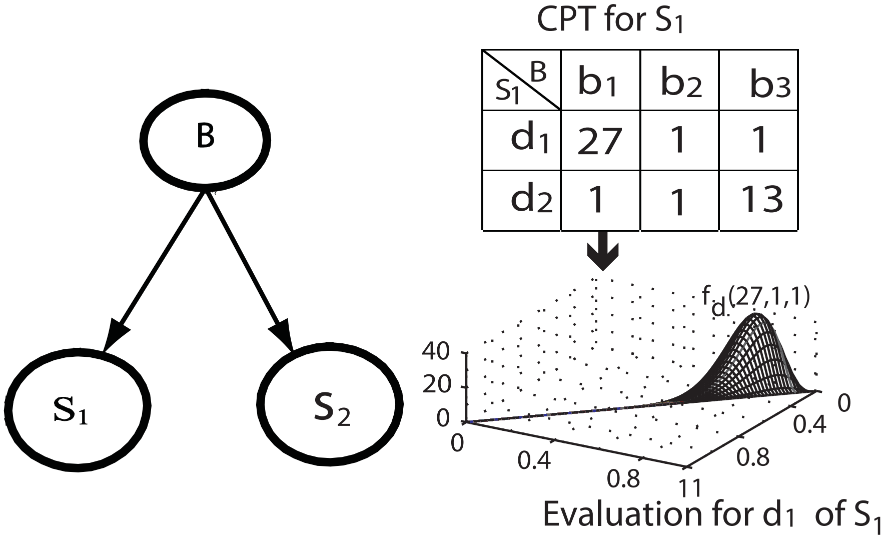
(1) Receive $D[t]$ and assign α_j as the number of observation of d_i

(2) Calculate degree of confidence $C_j[t]$

(3) Calculate change in the degree of confidence, E_j

(4) If $E_j \geq \theta$, then $D[t]$ is significant; $n_{ij}^{sig} := n_{ij}^{sig} + 1$ and go to Step 1

(5) Else $D[t]$ is insignificant; discards it and go to Step 1



S1 \ B	b1	b2	b3
d1	27	1	1
d2	1	1	13

Figure 4.2: Evaluation of significance and CPT for each sensor. Here $d_i \epsilon \{d_1, d_2\}$ and $b_i \epsilon \{b_1, b_2, b_3\}$.

This sensor proposition level significance evaluation process is illustrated in Fig. 4.2. The Fig. 4.2 shows a sample Bayesian network with CPT for one sensor (S_1) and the corresponding Dirichlet distribution for d_1 of s_1 for significance evaluation for a proposition (d_1). From Fig. 4.2 we can imagine how significance evaluation is done on each sensor proposition. Here, we explain the probability calculation based on the concepts explained above. Let n_{ij}^{sig} be the number of significant observations while the sensor proposition d_i is observed for b_j and N_j^{sig} be the number of significant observation of behavior b_j. Then we get

$$P\left(d_i|b_j\right) = P\left(S = d_i|B = b_j\right) = \frac{n_{ij}^{sig}}{N_j^{sig}}, \tag{4.1}$$

$$P\left(b_j\right) = \frac{N_j^{sig}}{\sum_j N_j^{sig}} \tag{4.2}$$

Now as discussed in chapter3 because of the significance evaluation N_2^{sig} can be almost equal to N_1^{sig}. Therefore, the prior probability remain unbiased. Also as significance evaluation is done on each sensor proposition the number of observation of sensor remained small. As a result the system could rapidly adapt to user policy by using handling both prior and conditoinal probability.

4.4 Experimental setup

We developed a teaching and learning system in a virtual environment that incorporated our concept. The environment as shown in Fig. 4.3 and Fig. 4.4 was prepared using Webot[70] real-time simulation software. The first experimental environment had an enclosed area of 8 [m]× 8 [m] with a static square obstacle of 1 [m]× 1 [m] placed inside the area. The second environment has an enclosed area of 8 [m]× 8 [m] and three moving obstacle (wondering robots) were placed in the square area. The user controls the robots with a lever joystick. In these experimental setups, user policy corresponds to avoid and approach. Avoid policy is accomplished by turning left when there is an obstacle on the right and vice versa. Approach behavior is accomplished by approaching the obstacle when there is one. Whether the obstacle is on the right or left side is determined by the laser distance sensor return value from sensors mounted on the front side of the robot. For the experiment we used a Bayesian network of eight distance sensor nodes and a behavior node as shown in Fig.4.5.

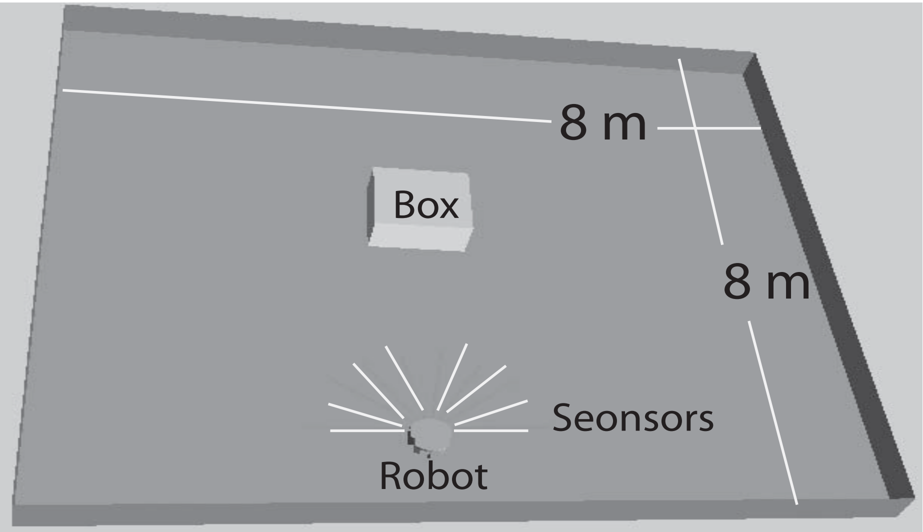


Figure 4.3: Virtual static experimental environment.

The robot model had eight front laser distance sensors (S_i,$i = 1, 2, \ldots, 8$) mounted on the front to measure the distance to obstacles along a horizontal distance to the obstacle. Joystick inputs were translated into discrete instructions by using a predetermined threshold. We found [103] that area of integration was inversely proportional and threshold value was directly proportional to the time required reach the evaluation criteria respectively. Therefore we set the area of integration to the maximum non-overlapping area and the threshold to 1.0×10^{-6}. These values for the area of integration and the threshold can be set according to the desired rapidity, and can vary from application to application.

The user can teleoperate the robot at any time. When user do not operate

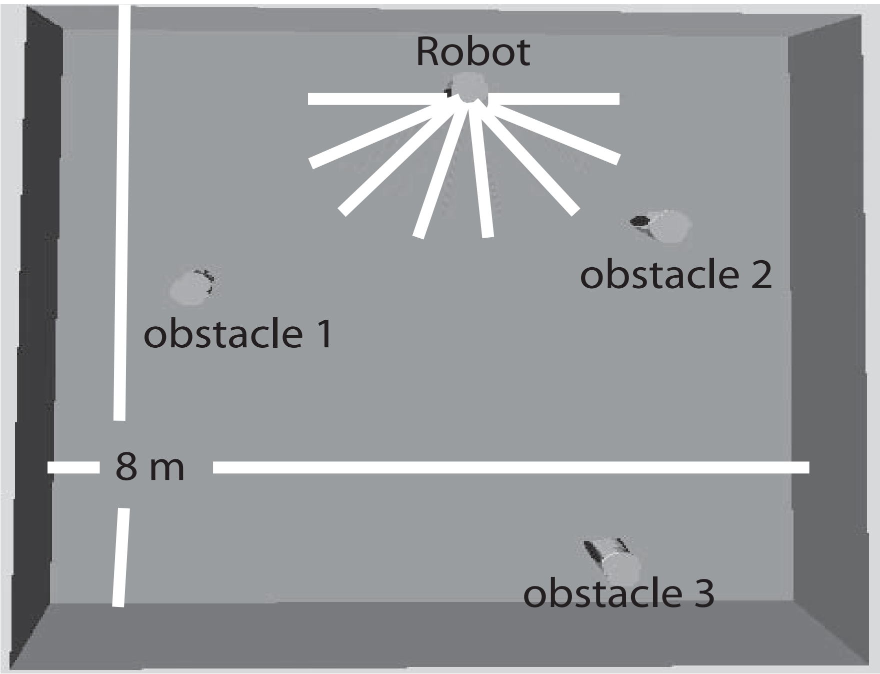


Figure 4.4: Virtual dynamic experimental environment.

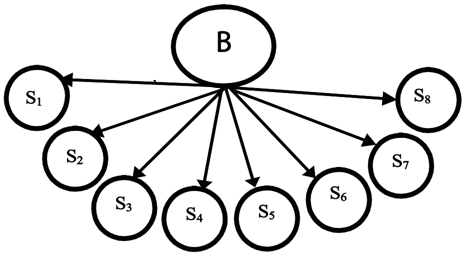


Figure 4.5: A Bayesian network used in the experiment.

the robot, it operates automatically with it's own degree of confidence in its behavior node. Three behaviors are taught in the experiment; go forward(b_1), turns left(b_2) and turn right(b_3). Previously we have shown that our algorithm can adapt to the user preference [105] by evaluating the significance of the behavior data. In the current experiments the user policy was changed from avoid to approach after a fixed amount of time (200 time steps). In our current work the evaluation of significance of the data is carried out fir each proposition of each sensor. In the following section the results of the significance evaluation in both static and dynamic environment is given.

4.5 Experimental results

4.5.1 For static environment

In the experiment the robot was to avoid in a few trial runs, and then policy was changed to approach. Fig. 4.6 shows a snapshot of the CPT for sensor number 5. The number of data points evaluated as significant and kept in secondary database for that particular sensor. The numbers in the CPT shows only the significant data is stored in the secondary database for each sensor proposition. Flat part of the graph represents times when data is evaluated as insignificant and discarded or when the robot operated automatically. From the CPT we can also see that the policy was overridden by accepting data for different sensor propositions. Policy adaptation is accomplished through the integration of all significant data for each sensor proposition.

Fig. 4.7 shows the changes in probability of the degree of confidence during teleoperation. This is an integrated probability over all sensor elements. Robot could rapidly adapted to the new policy around 320 by overriding the previous policy.

initial CPT

S\B	b_1	b_2	b_3
d_1	1	1	1
d_2	1	1	1

CPT when Avoid

S\B	b_1	b_2	b_3
d_1	28	8	1
d_2	1	26	1

CPT when Approach

S\B	b_1	b_2	b_3
d_1	33	8	1
d_2	39	28	1

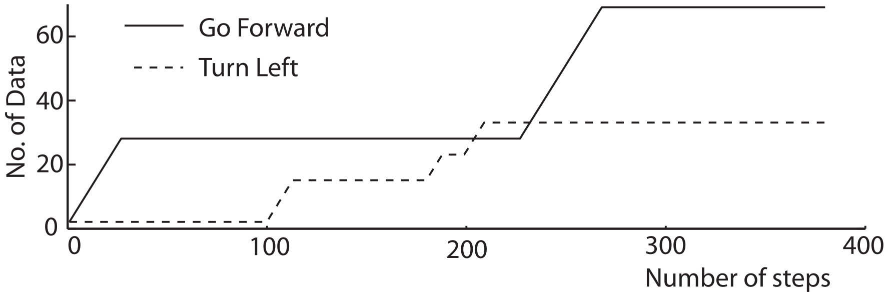


Figure 4.6: CPT and Number of data in the secondary database during policy adaptation for sensor node 5.

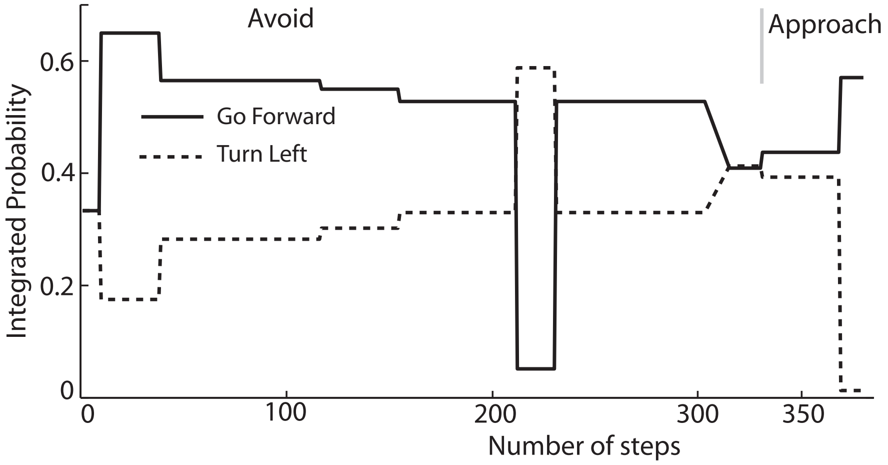


Figure 4.7: Integrated probability of behavior during policy adaptation.

4.5.2 For dynamic environment

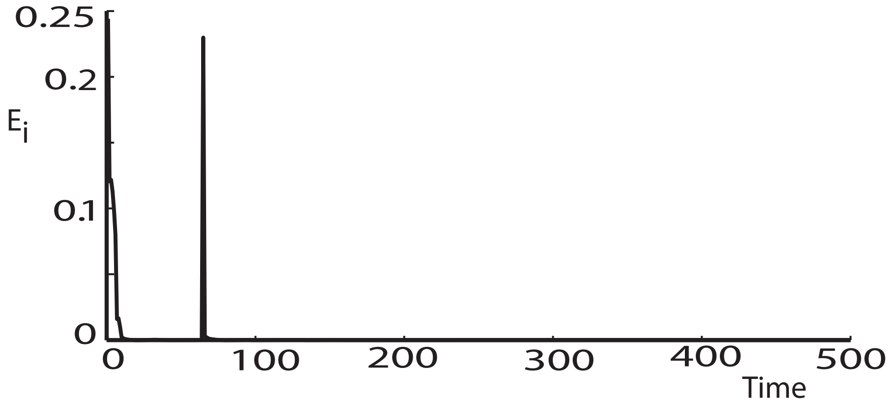


Figure 4.8: Change in degree of confidence of the proposition near ($d_1 = 0$) for sensor number 5.

Fig.4.8 shows the change in the degree of confidence for sensor proposition $d_1 = 0$ of sensor number 5. The figure confirms that the change in the degree of confidences decreases for the same behavior and increases for the a different behavior. Data evaluated as significant were kept in a secondary database for that particular sensor. The conditional probability table is updated only with significant data for each sensor proposition. Policy adaptation was accomplished through the integration of all significant data for each sensor proposition. Fig.4.9 and Fig.4.10 shows the robot's avoid and approach behavior, respectively after adaptation (time is given is seconds).

Fig.4.11 shows the change in the probability for different user behaviors. This shows the integrated probability over all sensor elements. The robot rapidly adapted to a new policy during the operation by overriding the previous policy twice. In this case rapidness is 0.66. As policy is a mapping from observation to action, it takes a few trials to learn or adapt to new policy and in that regard rapidness value of 0.66 is promising.

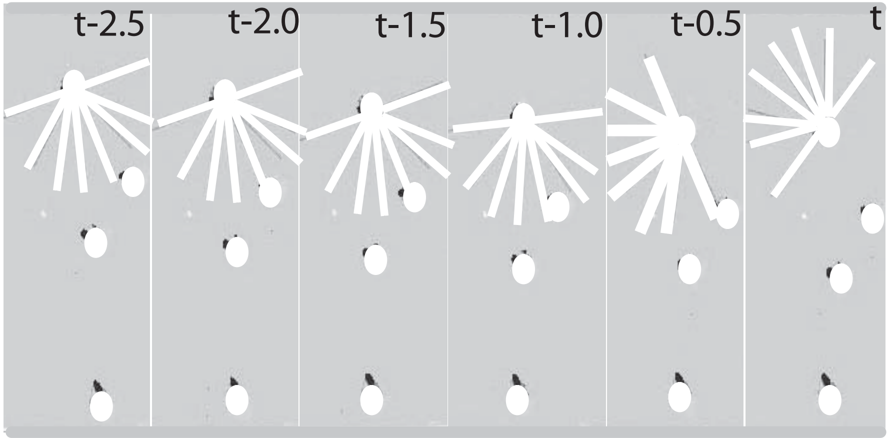


Figure 4.9: Robot behavior when the policy was to avoid.

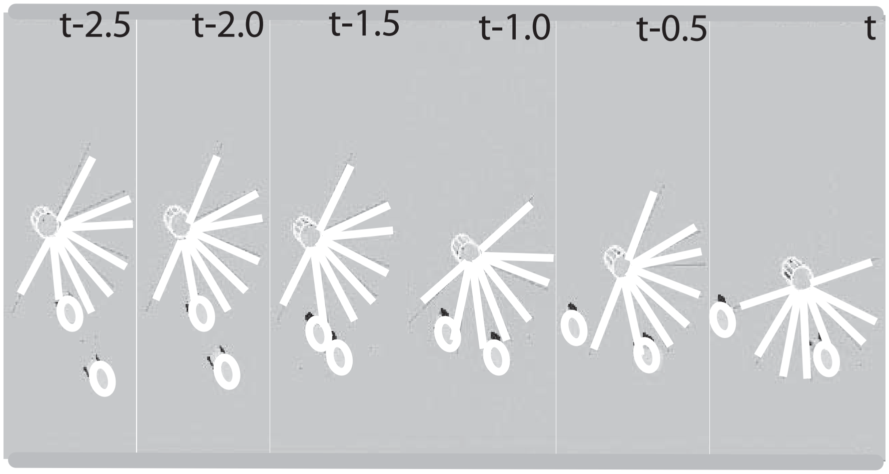


Figure 4.10: Robot behavior when the policy was to approach.

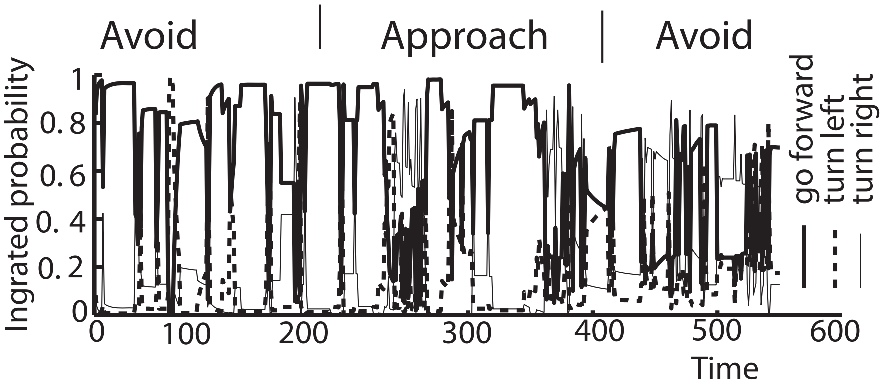


Figure 4.11: Integrated probability of behavior during policy adaptation.

4.6 Conclusion

The experimental results showed that our method could adapt to a user's policy on the basis of a significance evaluation using the change in degree of confidence. We described the significance evaluation for each sensor proposition of each sensor. In [103], we showed that a robot could adapt to a new user preference by evaluating the significance of behavior data. Robot could use only the prior probability as per Eq(4.2) and hence could adapt to the user's preference. The problem was that the robot could not rapidly adapt to user policy. In this chapter, the problem was solved through the use of both conditional probabilities Eq(4.1) and prior probabilities Eq(4.2). In this chapter we reported the experimental results in both static and dynamic environments. As for rapidness evaluation we used the same significance evaluation criteria for each sensor proposition. Hence the number of data in database will remain small. As a result time for adaptation to a new policy will remain constant for particular choice of θ and δ no matter for how long the robot is learning one policy. It will change with frequency of observation as we discussed before. Bayesian network that adapt without our significance evaluation method requires many trials thus making rapidness very small. The experimental results confirmed that only significant sensor observations are enough for rapid policy adaptation. In the next chapter we would like to integrate interactive communication with rapid policy adaptation method.

Chapter 5

Conclusion and future work

We use several simulated and real-world experimental domains to evaluate the algorithms presented in this thesis. A complete discussion of each algorithm is included with each chapter. The following sections presents contribution of this research, limitation of the presented algorithms and future works.

5.1 Contribution

The main contributions of the thesis are: **A novel method for rapid preference adaptation**. We show that the method of significance evaluation can be used to handle prior probability for rapid preference adaptation. Change in degree of confidence based significance evaluation method is used to select important action so that prior information become uninformative. It made rapid preference adaption possible. We demonstrate the utility of rapid preference adaptation method in the mobile robot context. We show that robot can automatically adapt rapidly to user preference when significance evaluation is used.

A novel method for rapid policy adaptation. We show that the method of significance evaluation can be used for rapid policy adaptation. Significance evaluation method is used to select important observation for addressing both prior and conditional probability. We show that robot can automatically adapt rapidly to user policies when significance evaluation is used.

Two techniques for representing and teaching collaborative behavior using demonstration. These techniques are based on different information sharing strategies: implicit coordination and coordination through active communication. We are now working on interactive communication that can be used for handling unstable confidence for rapid policy adaptation. Preliminary result of this work is presented in the conferences [104].

5.2 Limitation

There are few limitations of our work. We have chosen θ and Δ empirically. But in the ideal case the values of θ and Δ should be determined automatically. Sensor observation data has been discretized using a predefined threshold. But this discretization of sensor observation data should also be done automatically in ideal case. We have used a predefined Bayesian network structure. But in ideal case the network structure should be learned dynamically using observation.

5.3 Future Work

I have started my work with rapid policy adaptation for human centered robot. I would like contribute toward a solution for rapid adaptation in future. Here I am discussing about my ongoing work and a few possible work that might be undertaken in near future.

5.3.1 Adaptation based on interactive communication

Recent interest in intelligent robots represents building complex systems that is capable of holding belief in the state of the world. They come to hold these beliefs through existing data and by deriving new belief from interaction with user as a result of internal reasoning. Intelligent robotic systems are deemed to be agents capable of communicating about the events and confidence of the world which they share with their users.

In our previous works [107][108] we have shown that *rapid* behavior adaptation are possible based on the change in the degree of confidence based significance evaluation of observation and action. In those works we assumed that data is always available and user demonstrate until the robot has high confidence. During learning progression from low confidence to high confidence the robot simply followed user action of those methods. That system is practical if the user is infinitely patient. But we can expect human to be lazy and only wish to respond if necessary to improve the system's behavior or teach a new task. In this chapter we introduced interactive teaching in which robot will present its internal state as confidence and ask for user to teach based on the confidence. This confidence value can be used in many ways. Firstly, it can be used to recognize when the system has adequately learned a task. When the confidence value associated with queries is high, it means that the system has enough information to make a good prediction of the appropriate output. Teaching may then cease and the platform can proceed to act autonomously. Conversely, when the confidence value falls, it means that the learning algorithm is operating in space that it is unfamiliar with and perhaps its predicted outputs should not be trusted. In this case, signals can be sent to the user requesting more teaching (giving initiative to the teacher). In this work we integrate our policy learning method and interactive communication. Together, rapid adaptation and interactive communication algorithm form an interactive learning in which the learner and user teacher play collaborative role.

Interactive communication is a part of the policy learning algorithm in which the agent must select data, as it interacts with the human. At each time step, the algorithm uses confidence thresholds to determine whether a user action in the agents current state will provide useful information and improve the agents policy. If user action is required, the agent requests to the teacher, and updates its policy based on the resulting action. Otherwise the agent continues to perform its task autonomously based on its policy.

There are two distinct situations in which the agent requires help from the user, *unfamiliar* states and *ambiguous* states. An unfamiliar state occurs when the agent encounters a situation that is significantly different from any previously demonstrated state. Ambiguous states occur when the agent is unable to select between multiple actions with certainty. This situation can result when teaching of different actions from similar states make accurate classification impossible. In these cases, additional teaching may help to disambiguate the situation. The goal of the interactive adaptation algorithm is to update CPT for high confidence (autonomous execution) and low confidence (interactive teaching) in a way such that unfamiliar and ambiguous cases fall into the low confidence areas.

Here we can give tow interaction scenario for unfamiliar states and ambitious state. Unfamiliar situation might start at the beginning of the process when it start to evaluate significance of data and update CPT. Scenario for interaction during this unfamiliar state is as follows:

- **robot :** *unfamiliar* state, do you want learn with this data enter *yes* or*no* for +/- confirmation
- **user :** *yes*

In such interaction the robot will update CPT with the same observation until that data become insignificant. Therefore the robot need to clarify whether the user really want take this action because the robot has a high belief to follow the previous policy. A *no* confirmation by the user is considered as false positive data and will be ignored. Scenario for such interaction during this ambiguous state is as follows:

- **robot :** *ambiguous* state, do you really want this action; enter *yes* or*no* for +/- confirmation
- **user :** *yes*

In such case the robot will execute the user action. Policy adaptation can be accomplished through the significance evaluation and interactive teaching.

5.3.2 Bayesian network structure

In our current work we have predefined the Bayesian network structure. In ideal case the the system should learn Bayesian network online using observation. There are two types of algorithms for Bayesian structure learning is useful to accomplish that task namely constraint based algorithms and scoring based algorithms. It is possible to learn the Bayesian Network structure by identifying the conditional independence relationships among the nodes. Using some statistical tests (such as chi-squared or mutual information), we can find the conditional independence relationships among the nodes and use these relationships as constraints to construct a Bayesian Network. These algorithms are referred as dependency analysis based algorithms or constraint-based algorithms [83][74]. An alternative method of structural learning uses optimization based search. It requires a scoring function and a search strategy. A common scoring function is posterior probability of the structure given the training data. The time requirement of an exhaustive search returning back a structure that maximizes the score is super-exponential in the number of variables. A local search strategy makes incremental changes aimed at improving the score of the structure. A global search algorithm like Markov Chain Monte Carlo can avoid getting trapped in local minima. Therefore

I would like to work on online structure learning and use the structure with our algorithm.

5.3.3 Adapting threshold automatically

In this study we had empirically determined two values and we need to consider how to automate those issues as well. The threshold value for which change in the degree of confidence is determined for data significance and high confidence area of the distribution for integration. Although these values will depend on application, we need to explore if we can automate these values in those application domain.

5.3.4 Discretization of observation

In our current work discretization of sensor observation is done in predefined manner. But discretization should be determined on-line. The chi-square-based criteria [56] [57] [16] focus on the statistical point of view whereas the entropy-based criteria [21][86] focus on the information theoretical point of view. Other criteria such as Gini [18] try to find a trade off between information and statistical properties. I would like to work in near future on how to get the correct discretized states from observation.

5.4 Summary

This thesis has developed the concept of significance evaluation of observation data. We have shown that significance evaluation of observation data can be used for rapid behavior adaptation for human centered robot. Our algorithm for rapid behavior adaptation can deal with both prior and conditional probability, can detect the change point automatically, and is amenable to online robotic behavior adaptation. We have verified the usability of the algorithm on simulated environment and on real environment. This thesis provide a general and extensible approach for rapid behavior adaptation.

Bibliography

[1] P. Abbeel and A. Y. Ng. Apprenticeship learning via inverse reinforcement learning. In **Proceedings of the 21st International Conference on Machine Learning (ICML '04), Banff, Canada**, pages 1–8, 2004.

[2] A.Chella, E. Pagello, E. Menegatti, R.Sorbello, S. M. Anzalone, F. Cinquegrani, L. Tonin, F.Piccione, K.Prifitis, C. Blanda, E. Buttita, and E. Tranchina. A bci teleoperated museum robotic guide. In **International Conference on Complex, Intelligent and Software Intensive Systems, Fukuoka, Japan**, pages 783–788, 2009.

[3] R. Aler, O. Garcia, and J. M. Valls. Correcting and improving imitation models of humans for robosoccer agents. **Evolutionary Computation**, 3(2-5):2402–2409, 2005.

[4] A. Alissandrakis, C. L. Nehaniv, and Kerstin Dautenhahn. Towards robot cultures? **Interaction Studies: Social Behaviour and Communication in Biological and Artificial Systems**, 5(1):3–44, 2004.

[5] B. Argall, B. Browning, and M. Veloso. Learning to select state machines on an autonomous robot using expert advice. In **In Proceedings of the IEEE International Conference on Robotics and Automation, Rome, Italy**, pages 2124–2129, 2007.

[6] B. Argall, B. Browning, and M. Veloso. Learning robot motion control with demonstration and advice-operators. In **In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Nice, France**, pages 399–404, 2008.

[7] C. G. Atkeson and S. Schaal. Robot learning from demonstration. In **International Conference on Machine Learning, San Francisco, USA**, pages 12–20, 1997.

[8] F. Aznar, M. Sempere, M. Pujol, and R. Rizo. A cognitive model for autonomous agents based on bayesian programming. **Brain, Vision, and Artificial Intelligence, Lecture Notes in Computer Science**, 3704/2005:277–287, 2005.

[9] D. C. Bentivegna, C. G. Atkeson, and G. Cheng. Learning from observation and practice using primitives. In **AAAI Fall Symposium Series, Symposium on Real-life Reinforcement Learning, Washington,USA**, 2004.

[10] D. C. Bentivegna, C. G. Atkeson, A. Ude, and Gordon Cheng. Learning to act from observation and practice. **International Journal of Humanoid Robotics**, 1(4):585–611, 2004.

[11] A. Bifet and R. Gavalda. Learning from time-changing data with adaptive windowing. In **SIAM International Conference on Data Mining, Minnesota, USA**, 2007.

[12] A. Billard, S. Calinon Y. Epars, G. Cheng, and S. Schaal. Discovering optimal imitation strategies. **Robotics & Autonomous Systems, Special Issue: Robot Learning from Demonstration**, 47(2):69–77, 2004.

[13] D. Billsus and M. J. Pazzani. User modeling for adaptive news access. **User Modeling and User-Adapted Interaction**, 10:147–180, 2000.

[14] A. Blum and P. Langley. Selection of relevant features and examples in machine learning. **Artificial Intelligence**, 97(1-2):245–271, 1997.

[15] J. D. Boskovic, J. Redding, and R. K. Mehra. Integrated health monitoring and adaptive reconfigurable control. In **Proceedings of Guidance Navigation and Control Conference**, 2007.

[16] M. Boulle. Modl: A bayes optimal discretization method for continuous attributes. **Machine Learning**, 65:131–165, 2006.

[17] C. Breazeal, G. Hoffman, and A. Lockerd. Teaching and working with robots as a collaboration. In **Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems, New York, USA**, pages 1030–1037, 2004.

[18] R. A. Olshen Breiman, J. H. Friedman and C. J. Stone. **Classification and Regression Trees**. Wadsworth International, 1984.

[19] B. Browning, L. Xu, and M. Veloso. Skill acquisition and use for a dynamically-balancing soccer robot. In **Proceedings of 19th National Conference on Artificial Intelligence (AAAI '04), San Jose, CA, USA**, pages 599–604, 2004.

[20] S. Calinon and A. Billard. Incremental learning of gestures by imitation in a humanoid robot. In **Second Annual Conference on Human-Robot Interactions, Arlington, VA, USA**, pages 255–262, 2007.

[21] J. Catlett. On changing continuous attributes into ordered discrete attributes. In **Proceedings of the European working session on learning on Machine learning, Porto, Portugal**, pages 164–178, 1991.

[22] J. Chen and A. Zelinsky. Programing by demonstration: Coping with suboptimal teaching actions. **The International Journal of Robotics Research**, 22(5):299–319, 2003.

[23] Y. Chen, X. Dang, H. Peng, and H. Bart. Outlier detection with the kernelized spatial depth function. **IEEE Transaction on Pattern Analysis and Machine Intelligence**, 31(2):288–305, 2009.

[24] S. Chernova and M. Veloso. Confidence-based policy learning from demonstration using gaussian mixture models. In **Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems, Hawii, USA**, pages 233:1–233:8, 2007.

[25] S. Chernova and M. Veloso. Multi-thresholded approach to demonstration selection for interactive robot learing. In **Proceeding of the third ACM/IEEE International Conference on Human-Robot Interaction (HRI '08), Amsterdam, The Netherlands**, pages 225–232, 2008.

[26] S. Chernova and M. Veloso. Interactive policy learning thorugh confidence-based autonomy. **Journal of Artificial Intelligence Research**, 34:1–25, 2009.

[27] P. Chiu and G. Webb. Using decision tree for agent modeling; improving prediction performance. **User Modeling and User-Adapted Interaction**, 8:131–152, 1998.

[28] J. A. Clouse. **On integrating apprentice learning and reinforcement learning**. PhD thesis, University of Massachusetts, Department of Computer Science, 1996.

[29] D. Cohn, L. Atlas, and R. Ladner. Improving generalization with active learning. **Machine Learning**, 15(2):201–221, 1994.

[30] M. Datar, A. Gionis, P. Indyk, and R. Motwani. Maintaining stream statistics over sliding windows. **SIAM Journal of Computing**, 31(6):1794–1813, 2002.

[31] K. Dautenhahn and C. Nehaniv. Mapping between dissimilar bodies: Affordances and the algebraic foundations of imitation. In **Second Conference on Autonomous Agents: Workshop on Agents in Interaction - Acquiring Competence through Imitation, Edinburgh, Scotland**, pages 64–72, 1998.

[32] A. Dearden and Y. Demiris. Learning forward model for robots. In **Proceedings of the 19th International Joint Conference on Artificial Intelligence, Edinburgh, Scotland**, pages 1440–1445. IJCAI, 2005.

[33] R. Dillman. Teaching and learning robot task via observation of human performance. **Robotics and Autonomous Systems**, 47:109–116, 2004.

[34] K. R. Dixon and P. K. Khosla. Learning by observation with mobile robots: A computational approach. In **Proceedings of the IEEE International Conference on Robotics and Automation, New Orleans, USA**, pages 102–107, 2004.

[35] J. Elman. Finding structures in time. **Cognitive Science**, 14:179–192, 1990.

[36] S. Fleck, F. Busch, P. Biber, W. Straer, and H. Andreasson. Omnidirectional 3d modeling on a mobile robot using graph cuts. In **IEEE International Conference on Robotics and Automation, Barcelona, Spain**, pages 1748–1754, 2005.

[37] H. Friedrich and R. Dillmann. Robot programming based on a single demonstartion and user intensions. In **3rd European Workshop on Learning Robots at ECML, Crete, Greece**, 1995.

[38] J. Gama, P. Medas, G. Castillo, and P. Rodrigues. Learning with drift detection. **Lecture Notes in Computer Science**, 3171:66–112, 2004.

[39] A. Garland and N. Lesh. Learning hierarchical task models by demonstration. In **Technical Report TR2003-01, Mitsubishi Electric Research Laboratories**, 2003.

[40] D. H. Grollman and O. C. Jenkins. Dogged learning for robots. In **IEEE International Confeence on Robotics and Automation, Roma Italy**, pages 2483–2488, 2007.

[41] D. H. Grollman and O. C. Jenkins. Learning robot soccer skill from demonstration. In **6th IEEE International Conference on Development and Learning, London, UK**, pages 276–281, 2007.

[42] D. H. Grollman and O. C. Jenkins. Sparse incremental learing for interactive robot control policy estimation. In **Proceedings of the IEEE International Conference on Robotics and Automation, Pasadena, CA, USA**, pages 3315–3320, 2008.

[43] M Haruno, D. M. Wolpert, and M Kawato. Mosaic model for sensorimotor learning an control. **Neural Computing**, 13:2201–2220, 2001.

[44] V. J. Hodge and J. Austin. A survey on outlier detection methodologies. **Artificial Intelligence Review**, 22:85–128, 2004.

[45] G. Hovland, P. Sikka, and B. J. McCarragher. Skill acquisition from human demonstration using a hidden markov model. In **IEEE International Conference on Robotics and Automation, Michigan, USA**, pages 2706–2711, 1996.

[46] T. Inamura, M. Inaba, and H. Inoue. Acquisition of probabilistic decision model based on interactive teaching method. In **International Conference on Advanced Robotics, Tokyo, Japan**, pages 523–528, 1999.

[47] T. Inamura, M. Inaba, and H. Inoue. User adaptation of human-robot interaction model based on bayesian network and introspection of interaction experience. In **International Conference on Intelligent Robots and Systems, Takamatsu, Japan**, pages 2139–2144, 2000.

[48] T. Inamura, M. Inaba, and H. Inoue. Pexis: Probabilistic experience representation based adaptive interaction system for personal robots. **Systems and Computers in Japan**, 35(6):98–109, 2004.

[49] T. Inamura, Y. Nakamura, and I. Toshima. Embodied symbol emergence based on mimesis theory. **International Journal of Robotics Research**, 23(4):363–377, 2004.

[50] T. Inamura, H. Tanie, and Y. Nakamura. Keyframe extraction and decompression for time series data based on continuous hidden markov models. In **Proceedingd of International Conference on Intelligent Robots and Systems (IROS2003), Las Vegas, USA**, pages 1487–1492, 2003.

[51] T. Inamura, I. Toshima, and Y. Nakamura. Acquisition and embodiment of motion elements in closed mimesis loop. In **Proc. of Int'l Conf. on Robotics and Automation (ICRA2002), Washington, DC, USA**, pages 1539–1544, 2002.

[52] I. Infantino, A. Chella, H. Dzindo, and I. Macaluso. A posture sequence learning system for an anthropomorphic robotic hand. **Robotics and Autonomous Systems**, 42:143–152, 2004.

[53] R. A. Jacob, M. A. Jordan, S. J. Nowlan, and G. E. Hilton. Adapting mixture of local expert. **Neural Computation**, 3:79–87, 1991.

[54] O. C. Jenkins and M. J. Mataric. Performance-derived behavior vocabularies: Data-driven acqusition of skills from motion. **. International Journal of Humanoid Robotics**, 1(2):237–288, 2004.

[55] M. Johnson and Y. Demiris. Perceptual perspective taking and action recognition. **International Journal of Advanced Robotic Systems**, 2(4):301–308, 2005.

[56] V. G. Kass. An exploratory technique for investigating large quantities of categorical data. **Applied Statistics**, 29:2:119–127, 1980.

[57] R. Kerber. Chimerge: Discretization of numeric attributes. **10th Intl. Conf. on Artificial Intelligence**, pages 123–128, 1992.

[58] D. Kifer, S. Ben-David, and J. Gehrke. Detecting change in data streams. **30th VLDB Conference, Toronto, Canada**, pages 180–191, 2004.

[59] A. Kleiner, M dietl, and B Nebel. **Towards a life-long learning soccer agent**. Springer, 2003.

[60] R. Klinkenberg. Learning drifting concepts: Example selection vs. example weighting. **Intelligent Data Analysis**, 8(3):281–300, 2004.

[61] J. F. Kolen. **Exploring the computational capabilities of recurrent neural networks**. PhD thesis, The Ohio State University, 1994.

[62] Y. Kuniyoshi, M. Inaba, and H. Inoue. Learning by watching: Extracting reusable task knowledge from visual observation of human performance. **IEEE Transactions on Robotics and Automation**, 10:799–822, 1994.

[63] M. V. Lent and J. E. Laird. Learning procedural knowledge through observation. In **Proceedings of the 1st International Conference on Knowledge Capture, Victoria, Canada**, 2001.

[64] A. Lockerd and C. Breazeal. Tutelage and socially guided robot learning. In **IEEE/RSJ International Conference on Intelligent Robots and Systems, Cambridge, MA, USA**, pages 3475–3480, 2004.

[65] R. Maclin and J. W. Shavlik. Creating advice-taking reinforcement learners. **Machine Learning**, 22(1-3):251–281, 1996.

[66] M. Markou and S. Singh. Novelty detection: a review - part 1: statistical approaches. **Signal Processing**, 83:2481–2497, 2003.

[67] T. Matsui, N. Inuzuka, and H. Seki. Adapting to subsequent changes of environment by learning policy preconditions. **International Journal of Computer and Information Science**, 3:49–57, 2002.

[68] N. Matsunaga, C.S. Smith, T. Kanda, H. Ishiguro, and N. Hagita. Robot behavior adaptation for human-robot interaction based on policy gradient reinforcement learning. **Journal of the Robotics Society of Japan**, 24(7):820–829, 2006.

[69] L. A. Meeden. An incremental approach to developing intelligent neural network controllers for robots. **IEEE Transactions on Systems, Man, and Cybernetics**, 26(3):474–485, 1996.

[70] O. Michel. Webots: Professional mobile robot simulation. **Journal of Advanced Robotics Systems**, 1(1):39–42, 2004.

[71] A. W. Moore. Variable resolution dynamic programming: Efefficientlyearning action maps in multivariate real-valued state-spaces. In **Eighth International Workshop on Machine Learning, San Francisco, CA, USA**, 1991.

[72] R. Munos and A. W. Moore. Variable resolution discretization in optimal control. **Machine Learning**, 49:291–323, 2002.

[73] J. Nakanishi, J. Morimoto, G. Endo, G. Cheng, S. Schaal, and Mitsuo Kawato. Learning from demonstration and adaptation of biped locomotion. **Robotics and Autonomous Systems**, 47:79–91, 2004.

[74] R. E. Neapolitan. **Learning Bayesian Networks**. Prentice Hall, 2004.

[75] U. Nehmzow. **Experiments in Competence Acquisition for Autonomous Mobile Robots**. PhD thesis, University of Edinburgh, 1994.

[76] U. Nehmzow, O. Akanyeti, C.Weinrich, T. Kyriacou, and S. Billings. Robot programming by demonstration through system identification. In **Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '07)**, pages 801–806, 2007.

[77] A. Y. Ng, A. Coates, M. Diel, V. Ganapathi, J. Schulte, B. Tse, E. Berger, and E. Liang. Inverted autonomous helicopter flight via reinforcement learning. In **International Symposium on Experimental Robotics, Singapore**, 2004.

[78] M. N. Nicolescu. **A framework for learning from demonstration, generalization and practice in human-robot domains.** PhD thesis, University of Southern California, 2003.

[79] M. N. Nicolescu and M. J. Mataric. Learning and interacting in humanrobot domains. **Transaction on systems, Man and Cybernetics-Pat A: Suystems and Humans**, 31(5):419–430, 2001.

[80] M. Ogino, H. Toichi, Y. Yoshikawa, and M. Asada. Interaction rule learning with a human partner based on an imitation faculty with a simple visuo-motor mapping. **Robotics and Autonomous Systems, Special Issue on The Social Mechanisms of Robot Programming by Demonstration**, 54(5):414–418, 2006.

[81] E. Oliveira and L. Nunes. **Learning by exchanging Advice**. Springer, 2004.

[82] V. Papudesi. Integrating advice with reinforcement learning. Master's thesis, University of Texas at Arlington, 2002.

[83] J. Pearl. **Probabilistic reasoning in Intelligent system; network of plausible inference**. Morgan Kaufman, 1988.

[84] P. K. Pook and D. H. Ballard. Recognizing teleoperated manipulations. In **Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '93), Georgia, USA**, pages 578–585, 1993.

[85] B. Price and C. Boutilier. Accelerating reinforcement learning through implicit imitation. **Journal of Artificial Intelligence Research**, 19:569–629, 2003.

[86] J. R. Quinlan. Induction of decision trees. **Machine Learning**, 1::81–106, 1986.

[87] R. P. N. Rao, A. P. Shon, and A. N. Meltzof. **A bayesian model of imitation in infants and robots**, chapter 11. Cambridge University Press, Cambridge, UK, 2004.

[88] M. T. Rosenstein and A. G. Barto. **Supervised actor-critic reinforcement learning**, chapter 14. John Wiley & Sons, Inc., New York, NY, USA, 2004.

[89] D. E. Rumelhart and J. L. McClelland. **Parallel Distrbuted Processing**. MIT Press, Cambridge, MA, 1986.

[90] P. E. Rybski and M. Veloso. From cmdash05 to cmrobobits : Transitioning multi-agent research with aibos to the classroom. In **Procedings of the AAAI05 Mobile Robot Competition and Exhibition Workshop, National Conference on Artificial Intelligence, Pittsburgh, PA**, pages 53–60, 2005.

[91] P. E. Rybski, K. Yoon, J. Stolarz, and M. Veloso. Interactive robot task training through dialog and demonstration. In **Proceedings of the ACM/IEEE international conference on Human-robot interaction, Virginia, USA**, pages 49–56, 2007.

[92] J. Saunders, C. L. Nehaniv, and K. Dautenhahn. Teaching robots by moulding behavior and scaffolding the environment . In **ACM/SIGCHI-SIGART Conference on Human Robot Interaction, Utah, USA**, pages 118–125, 2006.

[93] S. Schaal. **Advances in neural information processing systems**, chapter learning from demonstration, pages 1040–1046. MIT press, 1997.

[94] W. D. Smart. **Making Reinforcement Learning Work on Real Robots**. PhD thesis, Department of Computer Science, Brown University, Providence, RI, 2002.

[95] W. D. Smart and L. P. Kaelbling. Practical reinforcement learning in continuous spaces. In **Proceedings of the Seventeenth International Conference on Machine Learning, San Francisco, CA, USA**, pages 903–910, 2000.

[96] W. D. Smart and L. P. Kaelbling. Effective reinforcement learning for mobile robots. In **IEEE International Conference on Robotics and Automaiton, St. Louis, USA**, pages 3404–3410, 2002.

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

[98] J. D. Sweeney and R. A. Grupen. A model of shared grasp affordances from demonstration. In **Proceedings of the IEEE-RAS International Conference on Humanoids Robots, Pittsburgh, PA , USA**, pages 27–35, 2007.

[99] Y. Takahashi, K. Hikita, and Minoru Asada. A hierarchical multi-module learning system based on self-interpretation of instructions by coach. In **RoboCup 2003: Robot Soccer World Cup VII**, pages 548–555, 2004.

[100] M. D. Tandale and J. Valasek. Fault-tolerant structured adaptive model inversion control. **Journal of Guidance, Control and Dynamics**, 29(3):635–642, 2006.

[101] J. Tani. Model-based learning for mobile robot navigation from the dynamical systems perspective. **IEEE Transactions on Systems, Man, and Cybernetics**, 26(3):421–436, 1996.

[102] J. Tani and S. Nolfi. Learning to perceive the world as articulated: An approach for hierarchical learning in sensorimotor systems. **Neural Network**, 12(7-8):1131–1141, 1999.

[103] S. M. Tareeq and T. Inamura. A sample discarding strategy for rapid adaptation to new situation for bayesian behavior learning. In **Proceedings of the IEEE International Conference on Robotics and Biomemtics**, pages 1950–1955, 2008.

[104] S. M. Tareeq and T. Inamura. Interactive behavior adaptation through dialogue based on bayesian network. In **2nd Asian Conference on Machine Learning**, 2010.

[105] S. M. Tareeq and T. Inamura. Management of experience data for rapid adaptation to new policies based on bayesian significance evaluation. In **Fifteenth International Symposium on Artificial Life and Robotics**, pages 126–129, 2010.

[106] S. M. Tareeq and T. Inamura. Management of experience data for rapid adaptation to new preferences based on bayesian significance evaluation. **Advanced Robotics**, (in press), 2010.

[107] S. M. Tareeq and T. Inamura. Rapid behavior adaptation for human-centered robots based on integration of primitive confidences on multi-sensor elements. In **International Conference on Advanced Mechatronics**, pages 271–276, 2010.

[108] S. M. Tareeq and T. Inamura. Rapid behavior adaptation for human-centered robots in a dynamic environment based on the integration of primitive confidences on multi-sensor elements. **Artificial Life and Robotics**, 15:515–521, 2010.

[109] A. L. Thomaz and C. Breazeal. Reinforcement learning with human teachers: Evidence of feedback and guidance with implications for learning performance. In **Twenty-First Conference on Artificial Intelligence (AAAI06), Boston, MA, USA**, pages 1000–1005, 2006.

[110] D. R. Thompson. Domain-guided novelty detection for autonomous exploration. In **Proceedings of the 21st international Joint Conference on Artificial Intelligence, San Francisco, CA, USA**, pages 1864–1869, 2009.

[111] J. Urzelai and D. Floreano. Evolutionary robotics: Coping with environmental change. In **Genetic and Evolutionary Computation Conference, Las Vegas, Nevada, USA**, pages 941–948, 2000.

[112] J. Urzelai and D. Floreano. Evolutionary robots with fast adaptive behavior in new environments. In **Proceedings of the Third International Conference on Evolvable Systems: From Biology to Hardware, London, UK**, pages 241–251, 2000.

[113] W. T. B. Uther and M. M. Veloso. Tree based discretization for continuous state space reinforcement learning. In **In Artificial Intelligence/Innovative Applications of Artificial Intelligence, Menlo Park, CA, USA**, pages 769–774, 1998.

[114] H. Veeraraghavan and M. Veloso. Teaching sequential tasks with repetition through demonstration (short paper). In **Proceedings of the International Conference on Autonomous Agents and Multiagent Systems, Estoril, Portugal**, pages 1357–1360, 2008.

[115] R. M. Voyles and P. K. Khosla. A multi-agent system for programming robotic agents by human demonstration. In **Proceedings of AI and Manufacturing Research Planning Workshop**, pages 184–190, 1998.

[116] T. J. Walsh, K. Subramanian, and M. L. Littman. Generalizing apprenticeship learning across hypothesis classes. In **Proceedings of the 27 th International Conference on Machine Learning, Haifa, Israel**, pages 1119–1126, 2010.

[117] G. Widmer and M. Kubat. Learning in the presence of concept drift and hidden contexts. **Machine Learning**, 23:69–101, 1996.

[118] K. Yamanishi, J. Takeuchi, and G. Williams. On-line unsupervised outlier detection using finite mixtures with discounting learning algorithms. **Data Mining and Knowledge Discovery**, 8(3):275–300, 2004.

Appendix A

Block Diagram of the System

Block Diagram of the System

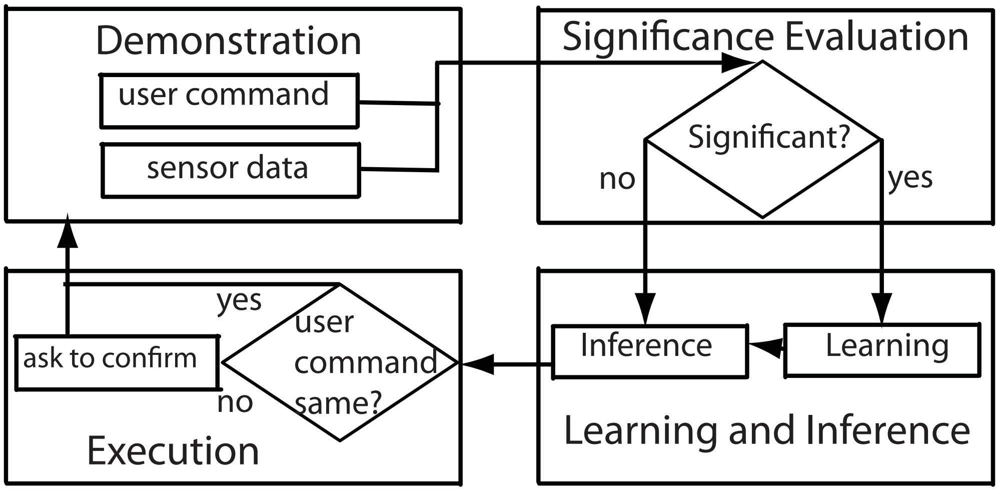


Figure A.1: The overall architecture of our system