Machine Learning, ?, 1–22 (1998)

Rapid Concept Learning for Mobile Robots

SRIDHAR MAHADEVAN, GEORGIOS THEOCHAROUS, NIKFAR KHALEELI

(mahadeva,theochar)@cps.msu.edu

Department of Computer Science, Michigan State University, East Lansing, MI 48864

Editor: Henry Hexmoor and Maja Mataric

Abstract. Concept learning in robotics is an extremely challenging problem. Sensory data is often high-dimensional, and noisy due to specularities and other irregularities. In this paper, we investigate two general strategies to speed up learning, based on spatial decomposition of the sensory representation, and simultaneous learning of multiple classes using a shared structure. We study two concept learning scenarios: a hallway navigation problem, where the robot has to induce feature detectors such as "opening" or "wall". The second task is recycling, where the robot has to learn to recognize objects, such as a "trash can". We use a common underlying function approximator in both studies in the form of a feedforward neural network, with several hundred input units and multiple output units. Despite the high degree of freedom afforded by such an approximator, we show the two strategies provide sufficient bias to achieve rapid learning. We provide detailed experimental studies on an actual mobile robot called PAVLOV to illustrate the effectiveness of this approach.

Keywords: Robot learning, concept learning, neural networks.

1. Introduction

Programming mobile robots to successfully operate in unstructured environments, including offices and homes, is tedious and difficult. Easing this programming burden seems necessary to realize many of the possible applications of mobile robot technology [7]. One promising avenue towards smarter and easier-to-program robots is to equip them with the ability to *learn* new concepts and behaviors. In particular, robots that have the capability of learning concepts could be programmed or instructed more readily than their non-learning counterparts. For example, a robot that could be trained to recognize landmarks, such as "doors" and "intersections", would enable a more flexible navigation system. Similarly, a recycling robot, which could be trained to find objects such as "trash cans" or "soda cans", could be adapted to new circumstances much more easily than non-learning robots (for example, new objects or containers could be easily accommodated by additional training).

Robot learning is currently an active area of research (e.g. see [5], [6], [9], [16]). Many different approaches to this problem are being investigated, ranging from supervised learning of concepts and behaviors, to learning behaviors from scalar feedback. While a detailed comparison of the different approaches to robot learning is beyond the scope of this paper (see [17]), it is arguable that in the short term, robots are going to be dependent on human trainers for much of their learning.

Specifically, a pragmatic approach to robot learning is one where a human designer provides the basic ingredients of the solution (e.g. the overall control architecture), with the missing components being filled in by additional training. Also, approaches involving considerable trial-and-error, such as reinforcement learning [25], are difficult to use in many circumstances, because they require long training times, or because they expose the robot to dangerous situations. For these reasons, we adopt the framework of supervised learning, where a human trainer provides the robot with labeled examples of the desired concept.

Supervised concept learning from labeled examples is probably the most well-studied form of learning [20]. Among the most successful approaches are decision trees [23] and neural networks [19]. Concept learning in robotics is an extremely challenging problem, for several reasons. Sensory data is often very high-dimensional (e.g. even a coarsely subsampled image can contain millions of pixels), noisy due to specularities and other irregularities, and typically data collection requires the robot to move to different parts of its environment. Under these conditions, it seems clear that some form of a priori knowledge or *bias* is necessary for robots to be able to successfully learn interesting concepts.

In this paper, we investigate two general approaches to bias sensory concept learning for mobile robots. The first is based on *spatial decomposition* of the sensor representation. The idea here is to partition a high-dimensional sensor representation, such as a local occupancy grid or a visual image, into multiple quadrants, and learn independently from each quadrant. The second form of bias investigated here is to learn multiple concepts using a shared representation. We investigate the effectiveness of these two approaches on two realistic tasks, navigation and recycling. Both these tasks are studied on a real robot called PAVLOV (see Figure 1). In both problems, we use a standardized function approximator, in the form of a feedforward neural net, to represent concepts, although we believe the bias strategies studied here would be applicable to other approximators (e.g. decision trees or instance-based methods).

In the navigation task, PAVLOV is required to traverse across an entire floor of the engineering building (see Figure 10). The navigational system uses a hybrid two-layered architecture, combining a probabilistic planning and execution layer with a reactive behavior-based layer. The planning layer requires the robot to map sensory values into high-level features, such as “doors” and “openings”. These observations are used in state estimation to localize the robot, and are critical to successful navigation despite noisy sensing and actions. We study how PAVLOV can be trained to recognize these features from local occupancy grid data. We also show that spatial decomposition and multiple category learning provide a relatively rapid training phase.

In the recycling task, PAVLOV is required to find items of trash (e.g. soda cans and other litter) and deposit them in a specified trash receptacle. The trash receptacles are color coded, to make recognition easier. Here, we study how PAVLOV can be trained to recognize and find trash receptacles from color images. The data is very high dimensional, but once again, spatial decomposition and multi-category learning are able to sufficiently constrain the hypothesis space to yield fast learning.

The rest of the paper is organized as follows. We begin in Section 2 by describing the two robotics tasks where we investigated sensory concept learning. Section 3 describes the two general forms of bias, decomposition and sharing, used to make the concept learning problem tractable. Section 4 describes the experimental results obtained on a real robot platform. Section 5 discusses the limitations of our approach and proposes some directions for further work. Section 6 discusses some related work. Finally, Section 7 summarizes the paper.

2. Two Example Tasks

We begin by describing the real robot testbed, followed by a discussion of two tasks involving learning sensory concepts from high-dimensional sensor data. The philosophy adopted in this work is that the human designer specifies most of the control architecture for solving the task, and the purpose of sensory concept learning is to fill in some details of the controller.

2.1. PAVLOV: A Real Robot

Figure 1 shows our robot PAVLOV [1], a Nomad 200 mobile robot base, which was used in the experiments described below. The sensors used on PAVLOV include 16 ultrasound sonar and infra-red (IR) sensors, arranged radially in a ring. Two sets of bumper switches are also provided. In addition, PAVLOV has a color camera and frame-grabber. Communication is provided using a wireless Ethernet system, although most of the experiments reported in this paper were run onboard the robot's Pentium processor.

Figure 1. The experiments were carried out on PAVLOV, a Nomad 200 platform.

2.2. Navigation

Robot navigation is a very well-studied topic [1]. However, it continues to be an active topic for research since there is much room for improvement in current systems. Navigation is challenging because it requires dealing with significant sensor and actuator errors (e.g. sonar is prone to numerous specular errors, and odometry is also unreliable due to wheel slippage, uneven floors, etc.).

We will be using a navigation system based on a probabilistic framework, formally called partially-observable Markov decision processes (POMDP's) [4], [13], [21]. This framework uses an explicit probabilistic model of actuator and sensor uncertainty, which allows a robot to maintain belief estimates of its location in its environment. The POMDP approach uses a state estimation procedure that takes into account both sensor and actuator uncertainty to determine the approximate location of the robot. This state estimation procedure is more powerful than traditional state estimators, such as Kalman filters [14], because it can represent discontinuous distributions, such as when the robot believes it could be in either a north-south corridor or an adjacent east-west corridor.

For state estimation using POMDP's, the robot must map the current sensor values into a few high level observations. In particular, in our system, the robot generates 4 observations (one for each direction). Each observation can be one of four possibilities: *door*, *wall*, *opening*, or *undefined*. These observations are generated from a local occupancy grid representation computed by integrating over multiple sonar readings.

Figure 2 illustrates the navigation system onboard PAVLOV, which combines a high level planner with a reactive layer. The route planner and execution system used is novel in that it uses a discrete-event probabilistic model, unlike previous approaches which use a discrete-time model. However, as the focus of this paper is on learning the feature detectors, we restrict the presentation here to explaining the use of feature detectors in state estimation, and refer the reader to other sources for details of the navigation system [11], [18].

The robot maintains at every step a *belief state*, which is a discrete probability distribution on the underlying state space (e.g. in our environment, the belief state is a 1200-dimensional vector). If the current state distribution is α_{prior}, the state distribution α_{post}, after the execution of an abstract action a, is given by [2]

$$\alpha_{post}(s) = \frac{1}{scale} \sum_{s' \in S | a \in A(s')} P(s|s', a)\alpha_{prior}(s'), \ \ \forall s \in \mathcal{S} \tag{1}$$

This updated state distribution now serves as α_{prior} when the state distribution is updated to α_{post}, after an abstract observation o

$$\alpha_{post}(s) = \frac{1}{scale} O(o|s)\alpha_{prior}(s), \ \ \forall s \in \mathcal{S} \tag{2}$$

In both updates, *scale* is a normalization constant that ensures that

$$\sum_{s \in S} \alpha_{post}(s) = 1$$

This is necessary since not every action is defined in every state (for example, the action *go-forward* is not defined in states where the robot is facing a wall).

2.3. Abstract Observations

In each state, the robot is able to make an abstract observation. This is facilitated through the modeling of four virtual sensors that can perceive features in the nominal directions front, left, back and right. Each sensor is capable of determining if a percept is a *wall*, an *opening*, a *door* or if it is *undefined*. An abstract observation is a combination of the percepts in each direction, and thus there are 256 possible abstract observations. The observation model specifies, for each state and action, the probability that a particular observation will be made.

Denote the set of virtual sensors by I and the set of features that sensor $i \in I$ can report on by $Q(i)$. The sensor model is specified by the probabilities $v_i(f|s)$ for all $i \in I$, $f \in Q(i)$, and $s \in \mathcal{S}$, encoding the sensor uncertainty. $v_i(f \mid s)$ is the probability with which sensor i reports feature f in state s. An observation o is the aggregate of the reports from each sensor. This is not explicitly represented. We calculate only the observation probability. Thus, if sensor i reports feature f, then

$$O(o|s) = \prod_{i \in I} v_i(f|s) \tag{3}$$

Given the state, this assumes sensor reports from different sensors are independent.

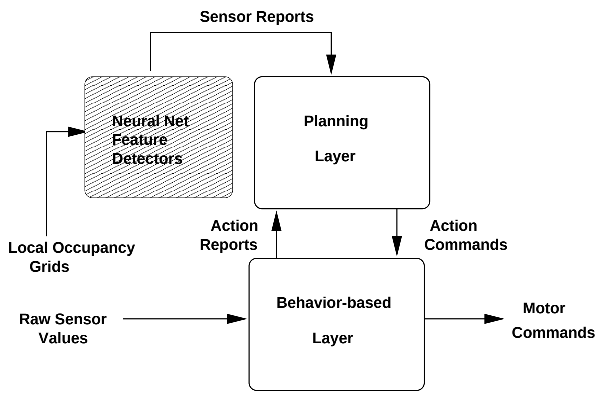

Figure 2. A hybrid declarative-reactive architecture for robot navigation. The neural net feature detectors (shaded box) are trained using spatial decomposition and multi-task learning.

2.4. Recycling

The second task we study is one where the robot has to find and pick up litter lying on the floor (e.g. soda cans and other junk) and deposit it in a colored trash receptacle (see Figure 3). This task involves several component abilities, such as locating and picking up the trash, and also subservient behaviors (such as avoiding obstacles etc.). However, for the purposes of this paper, we will mainly focus on the task of detecting a trash can from the current camera image, and moving the robot till it is located adjacent to the trash can.

Figure 3. Image of a trash can, which is color coded to facilitate recognition (this can is colored yellow).

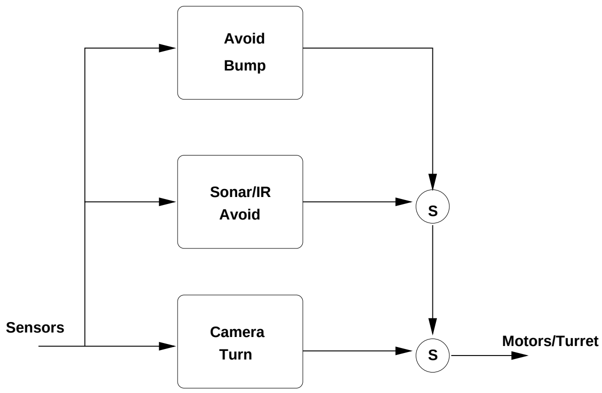

Figure 4. Behavior-based architecture for recycling task. The focus of sensory concept learning here is to improve "camera turn" behavior by learning how to detect and move towards trash cans.

The recycling task is accomplished using a behavior-based architecture [2], as illustrated in Figure 4. Only one of the behaviors, "camera turn" is improved by the sensory concept learning methods described here, in particular, by learning how to detect and move towards the trash can. The other behaviors implement a collection of obstacle avoidance algorithms, which are not learned.

3. Accelerating sensory concept learning

Learning sensory concepts is difficult because the data is often very high-dimensional and noisy. The number of instances is often also limited, since data collection requires running the robot around. In order to learn useful concepts, under these conditions, requires using some appropriate *bias* [20] to constrain the set of possible hypotheses [3] The study of bias is of paramount importance to machine learning, and some researchers have attempted a taxonomy of different type of bias (e.g. see [24]). Among the main categories of bias studied in machine learning are *hypothesis space bias* (which rules out certain hypotheses), and *preference bias* which ranks one hypotheses over another (e.g. prefer shallower decision trees over deeper ones).

In the context of robotics, the ALVINN system [22] for autonomous driving is a good example of the judicious use of *hypotheses bias* to speed convergence. Here, for every human provided example, a dozen or so synthetic examples are constructed by scaling and rotating the image input to the net, for which the desired output is computed using a known *pursuit* steering model. We present below two ways of accelerating sensory concept learning, which can also be viewed as a type of hypotheses space biases.

3.1. Spatial Decomposition

The sensory state space in both tasks described above is huge (of the order of several hundred real-valued inputs). The number of training examples available is quite limited, e.g. on the order of a few hundred at most. How is it possible to learn a complex function from such a large state space, with so little data? We use two general approaches to decompose the overall function learning problem.

The first idea is simple: partition the state into several distinct regions, and learn subfunctions for each region. The idea is illustrated in Figure 5. This idea is used in the navigation domain to train four separate feature detectors, one each for the front and back quadrants of the local occupancy grid, and one each for the left and right quadrants. There are two advantages of such a decomposition: each image generates four distinct training examples, and the input size is halved from the original input (e.g. in the navigation domain, the number of inputs is 512 rather than 1024).

3.2. Multi-class Learning

The second strategy used in our work to speed sensory concept learning is to learn multiple categories using a shared structure. This idea is fairly well-known in neural nets, where the tradeoff between using multiple single output neural nets vs. one multi-output neural net has been well studied. Work by Caruana [3] shows that even when the goal is to learn a single concept, it helps to use a multi-output net to learn related concepts. Figure 6 illustrates the basic idea. In the recycling domain, for example, the robot learns not just the concept of "trash can", but also whether the object is "near" or "far", on the "left" or on the "right". Simultaneously learning these related concepts results in better performance, as we will show below.

4. Experimental Results

The experiments described below were conducted over a period of several months on our real robot PAVLOV, either inside the laboratory (for recycling) or in the corridors (for navigation). We first present the results for the navigation task, and subsequently describe the results for the recycling task.

4.1. Learning Feature Detectors for Navigation

Given that the walls in our environment were fairly smooth, we found that sonars were prone to specular reflections in a majority of the environment. This made it difficult to create hard-coded feature detectors for recognizing sonar signatures. We show below that using an artificial neural network produced more accurate and consistent results. Not only was it easy to implement and train, but it is also possible to port it to other environments and add new features. Figure 7 shows the neural net used in feature detection. The net was trained using the quickprop method [8], an optimized variant of the backpropagation algorithm.

Sample local occupancy grids were collected by running the robot through the hallways. Each local occupancy grid was then used to produce 4 training patterns. The neural net was trained on 872 hand labeled examples. Since all sensors predict the same set of features, it was only necessary to learn one set of weights. Figure 8 shows the learning curve for the neural net, using batch update. Starting off with a set of random weights, the total error over all training examples converged to an acceptable range (< 1) within about 60 training epochs.

A separate set of data, with 380 labeled patterns, was used to test the net. This would be the approximately the number of examples encountered by the robot, as it navigated the loop in the Electrical Engineering department (nodes 3-4-5-6 in Figure 10). Feature prediction is accomplished by using the output with the maximum value. Out of the 380 test examples, the neural net correctly predicts features for 322, leading to an accuracy of 85%.

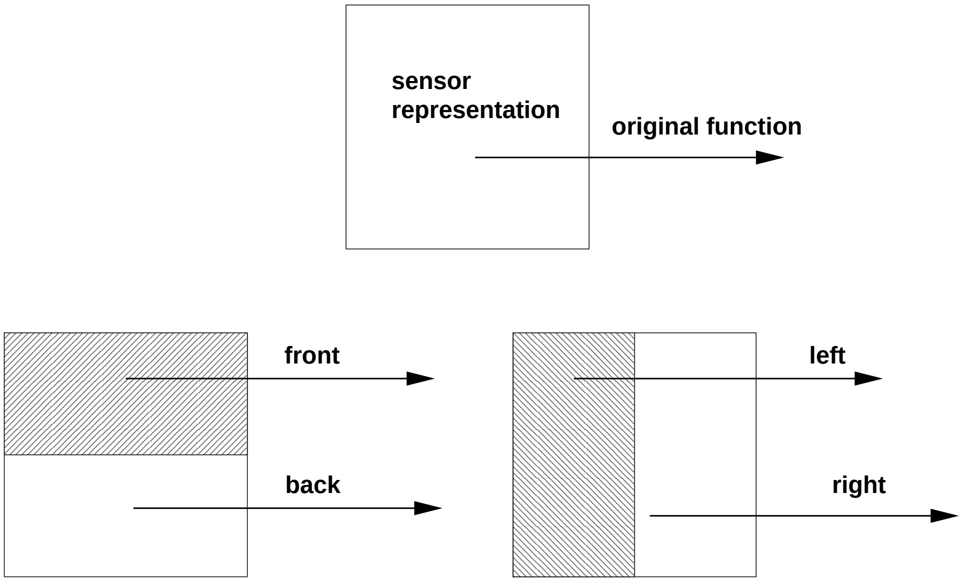

Figure 5. Spatial decomposition of the original sensory state helps speed learning sensory concepts. Here, the original sensory space is decomposed into a pair of two disjoint quadrants.

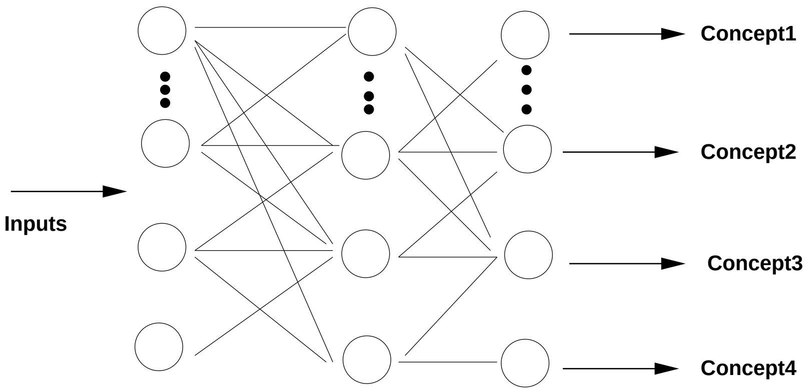

Figure 6. Learning multiple concepts simultaneously using a shared representation can speed sensory learning.

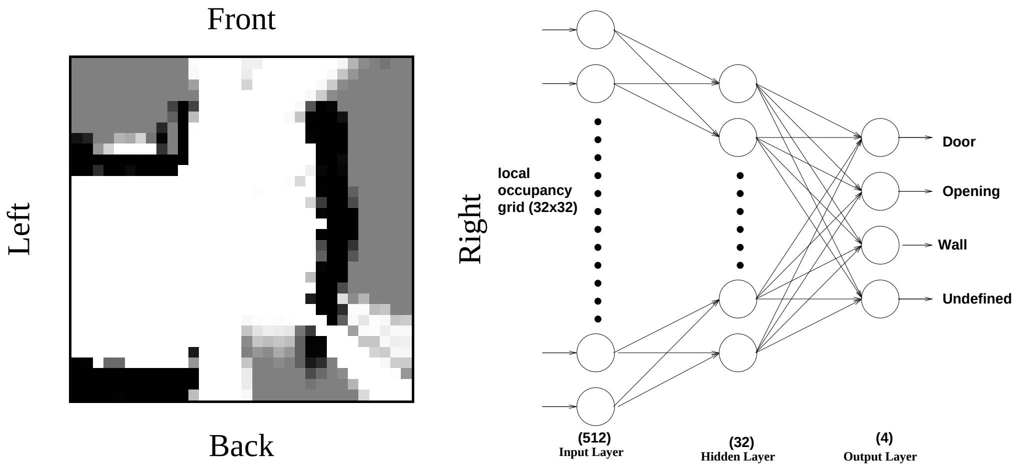

Figure 7. A local occupancy grid map, which is decomposed into four overlapping quadrants (left, right, top, bottom), each of which is input to a neural net feature detector. The output of the net is a multi-class label estimating the likelihood of each possible observation (door, opening, wall, or undefined). The net is trained on manually labeled real data.

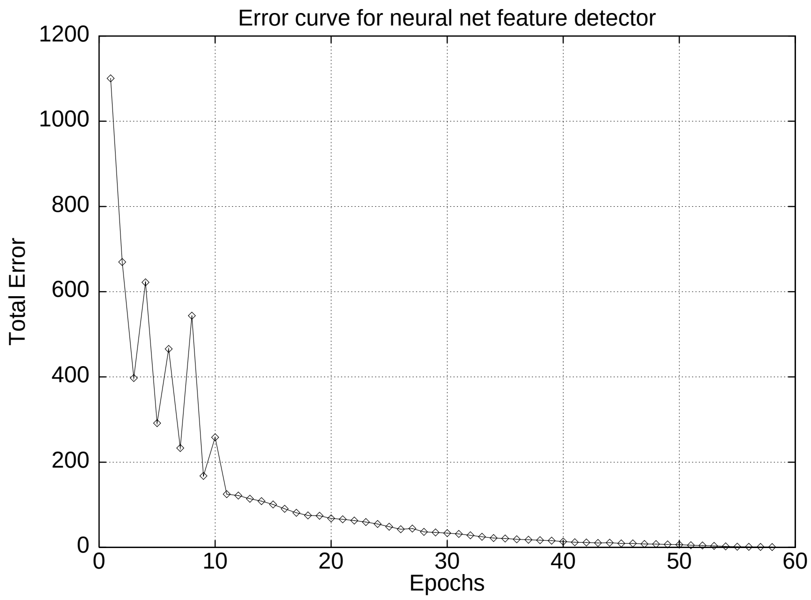

Figure 8. Learning curve for training neural net to recognize features. The net is trained on 872 hand labeled examples using quickprop.

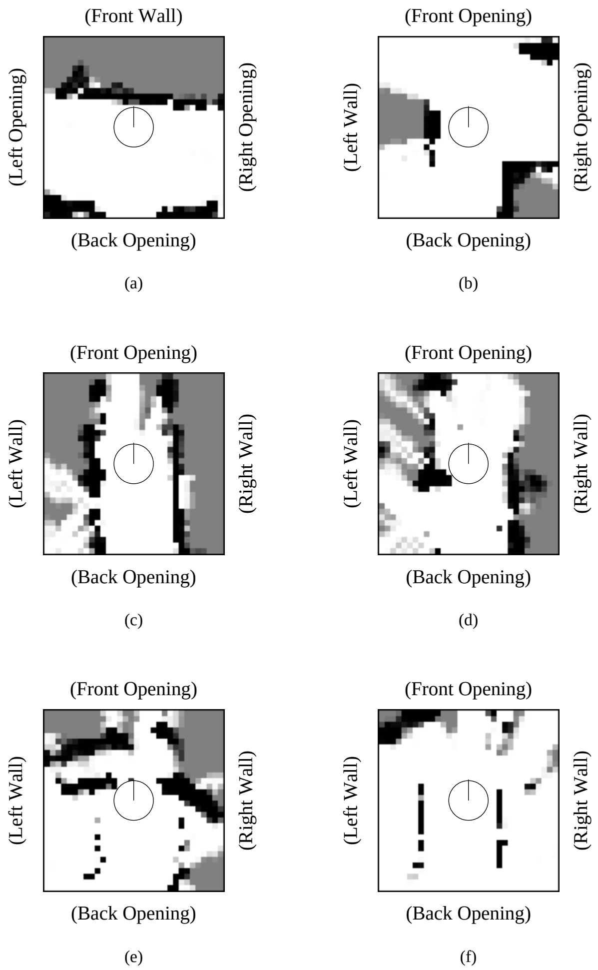

Figure 9. Sample local occupancy grids generated over an actual run, with observations output by the trained neural net. Despite significant sensor noise, the net is able to produce fairly reliable observations.

Figure 9 illustrates the variation in observation data, generated during an actual run. In these occupancy grids, free space is represented by white, while black represents occupied space. Gray areas indicate that occupancy information is unknown. The figures are labeled with the virtual sensors and corresponding features, as predicted by the neural net.

Specular reflections occur when a sonar pulse hits a smooth, flat surface angled obliquely to the transducer. The possibility exists that the sonar pulse will reflect away from the sensor, and undergo multiple reflections before it is received by the sensor. As a result, the sensor registers a range that is substantially larger than the actual range. In the occupancy grids, this results in a physically occupied region having a low occupancy probability. In Figure 9(a) where the specularities are relatively insignificant, the neural net does an accurate job of predicting the features. Effects of the specularities are noticeable in Figure 9(b) and Figure 9(c). In Figure 9(b) the neural net is able to predict a wall on the left, although it has been almost totally obscured by specular reflections. The occupancy grid in Figure 9(c) shows some bleed-through of the sonars. In both examples, the neural net correctly predicts the high level features. Figure 9(e) and Figure 9(f) are examples of occupancy grids where the effects of the specularities become very noticeable. In these examples specularities dominate, almost totally wiping out any useful information, yet the neural net is still able to correctly predict features.

From the presented examples, it is apparent that the neural net can robustly predict features in a highly specular environment. Testing the neural net on an unseen set of labeled data reveals that it is able to correctly predict 85% of the features. In addition, although examples have not been presented, the neural net is able to accurately predict features even when the robot is not approximately oriented along one of the allowed compass directions.

The navigation system was tested by running the robot over the entire floor of the engineering building over a period of several months (see Figure 10). The figure also shows an odometric trace of a particular navigation run, which demonstrates that despite significant odometric and sensor errors, the robot is still able to complete the task.

4.2. Learning to Find Trash Cans

We now present the experimental findings from the recycling task. In order to implement a similar neural network approach, we first took various snapshots of the trash can from different angles and distances using the on-board camera of PAVLOV. The images (100x100 color images) were labeled as to the distance and orientation of the trash can. Six boolean variables were used to label the images (front, left, right, far, near, very-near).

The inputs to our neural network were pre-processed selected pixels from the 100X100 images, and the outputs were the six boolean variables. The RGB values of the colored images were transformed into HSI values (Hue, Saturation, Intensity) which are better representatives of true color value because they are more invariant

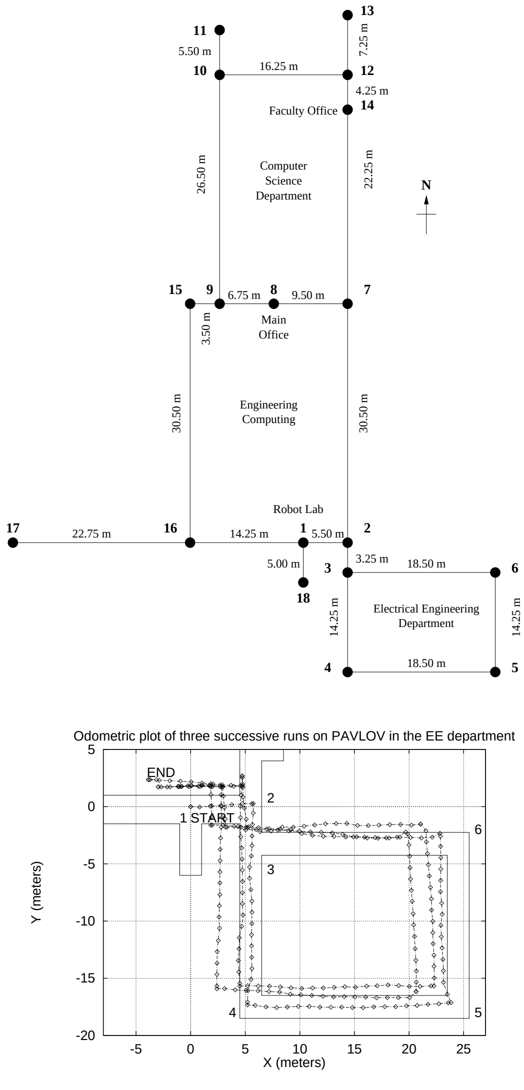

Figure 10. The 3rd floor of the engineering building was used to test the effectiveness of the feature detectors for navigation. The bottom figure shows an odometric trace of a run on PAVLOV, showing the robot starting at node 1, doing the loop (3-4-5-6), and returning to node 1. The robot repeated this task three times, and succeeded despite significant odometric errors.

to light variations [10]. Using an image processing program we identified the HSI values of the yellow color and based on those values we thresholded the images into black and white. We then sub-sampled the images into 400 pixels so that we could have a smaller network with far fewer inputs. The sub-sampling was done by selecting one pixel in every five.

Figure 11 shows the neural net architecture chosen for the recycling task. Figure 12 shows some sample images, with the output generated by the trained neural net. The neural net produces a six element vector as its output, with 3 bits indicating the direction of the trash can (left, front, or right), and 3 bits indicating the distance (far, near, very near). The figure shows only the output values that were close to 1. Note that the net can generate a combination of two categories (e.g. near and very-near), or even sometimes a contradictory labeling (e.g. far/near). In such cases, the camera turn behavior simply chooses one of the labels, and proceeds with capturing subsequent images, which will eventually resolve the situation (this is shown in the experiments below). Figure 13 shows the learning curve for training the trash can net.

Figure 14 shows the experimental setup used to test the effectiveness of the trash can finder. A single yellow colored trash can was placed in the lab at four different locations. In each case, the robot was started at the same location, and its route measured until it stopped adjacent to the trash can (and announced that it had found the trash can).[4] Figure 15 and Figure 16 show several sample trajectories of the robot as it tried to find the trash can. In all cases, the robot eventually finds the trash can, although it takes noticeably longer when the trash can is not directly observable from the starting position.

5. Limitations of the Approach

The results presented above suggest that high-dimensional sensory concepts can be learned from limited training examples, provided that a human designer carefully structures the overall learning task. This approach clearly has some definite strengths, as well as some key limitations.

- *Need for a teacher:* Supervised concept learning depends on a human teacher for providing labeled examples of the desired target concept. Previous work on systems such as ALVINN [22] has clearly demonstrated that there are interesting tasks where examples can be easily collected. Similarly, for the navigation and recycling task, we have found that collecting and labeling examples to be a fairly easy (although somewhat tedious) task. Nevertheless, this approach could not be easily used in domains where it is difficult for a human teacher to find a sufficiently diverse collection of examples.

- *Filling in details of a pre-specified architecture:* The approach taken in this paper assumes that the designer has already pre-specified much of the overall control structure for solving the problem. The purpose of learning is to complete

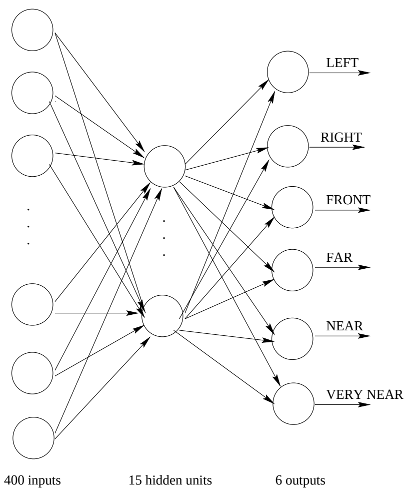

Figure 11. A neural net trained to detect trash cans.

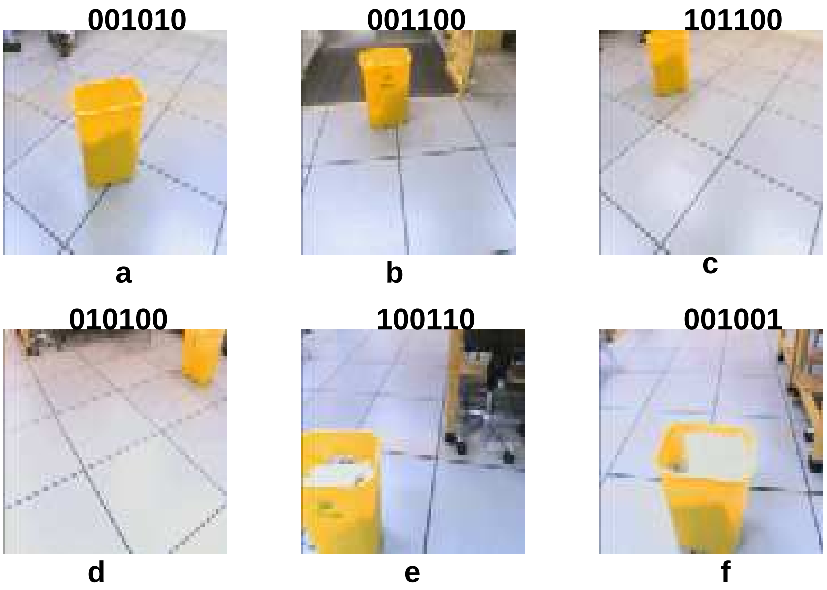

Figure 12. Sample images with the output labels generated by the neural net **a**: front, near. **b**: front, far. **c**: left, front, far. **d**: right, far. **e**: left, near, very-near. **f**: front, very-near.

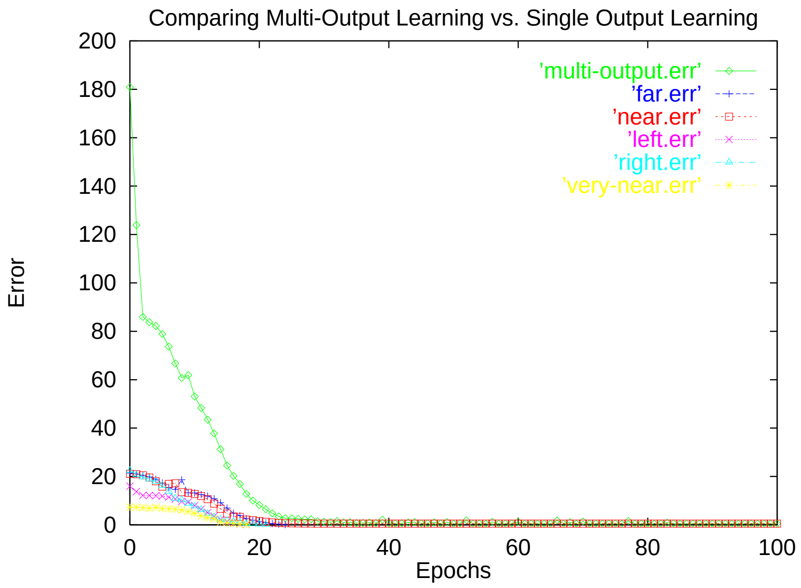

Figure 13. This graph compares the training time for a multi-output net vs. training a set of single output nets. Although the multi-output net is slower to converge, it performed better on the test data.

a few missing pieces of this solution. In the navigation task, for example, the feature detectors are all that is learned, since the overall planner, reactive behaviors, and state estimator are pre-programmed. Obviously, this places a somewhat large burden on the human designer.

- *Decomposable functions:* The sensory concepts being learned in the two tasks were decomposable in some interesting way (either the input or the output space could be partitioned). We believe many interesting concepts that robots need to learn have spatial regularity of some sort that can be exploited to facilitate learning.

6. Related Work

This research builds on a distinguished history of prior work on concept learning from examples, both in machine learning [20] and in robot learning [5], [9]. Here, we focus primarily on the latter work, and contrast some recent neural-net based approaches with decision-tree based studies.

ALVINN [22] uses a feedforward neural net to learn a steering behavior from labeled training examples, collected from actual human drivers. As noted earlier, ALVINN exploits a pursuit model of steering to synthesize new examples to speed

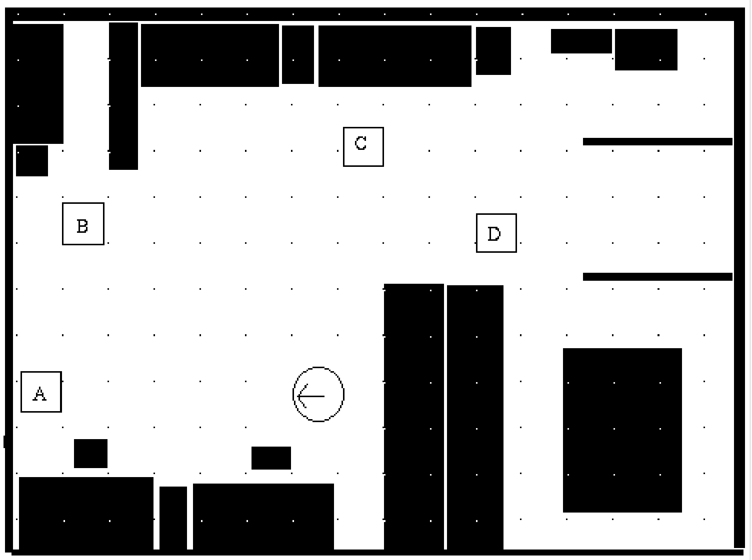

Figure 14. Environmental setup for finding trash cans.

learning. ALVINN differs from our work in that it directly learns a policy, whereas in our case the robot learns only feature detectors and recognizers. We believe that directly learning an entire policy is quite difficult, in general. In fact, in a subsequent across-the-country experiment, the direct policy learning approach was rejected in favor of a simpler feature-based approach similar to our work (except the templates were 1-dimensional rather than 2-dimensional, as in our work).

Thrun and Mitchell [27] propose a *lifelong learning* approach, which extends the supervised neural-net learning framework to handle transfer across related tasks. Their approach is based on finding *invariances* across related functions. For example, given the task of recognizing many objects using the same camera, invariances based on scaling, rotation, and image intensity can be exploited to speed up learning. Their work is complementary to ours, in that we are focusing on rapid within-task learning, and the invariants approach could be easily combined with the partioning and multi-class approach described here.

Such studies can be contrasted with those using decision trees. For example, Tan [26] developed an ID3 decision-tree based algorithm for learning strategies for picking up objects, based on perceived geometric attributes of the object, such as its height and shape. Salganicoff et al. [15] extended the decision-tree approach for learning grasping to an active learning context, where the robotic system could itself acquire new examples through exploration.

In general, the decision tree approaches seem more applicable when the data is not high-dimensional (in both the system just cited, the number of input dimensions is

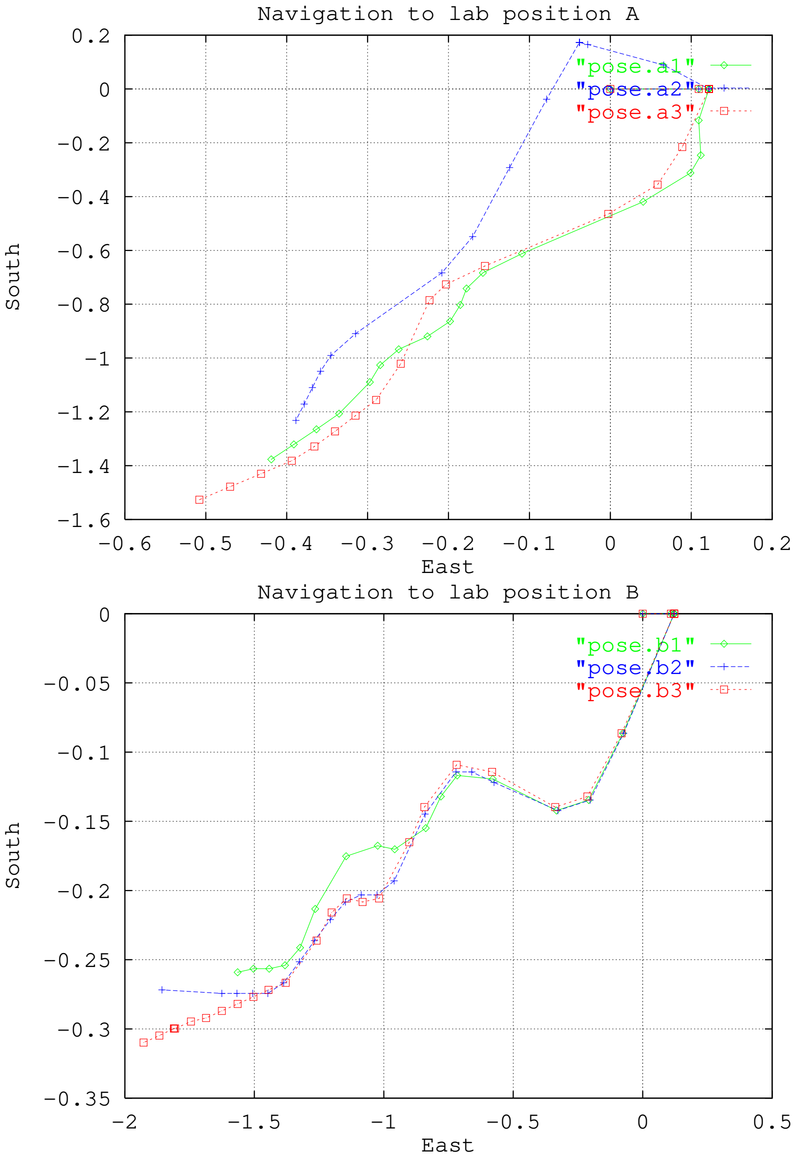

Figure 15. Three successful traces (starting at 0,0) of the robot navigating to the trash can, placed in positions A and B. In both positions A and B, the trash can was directly observable from the robot starting position.

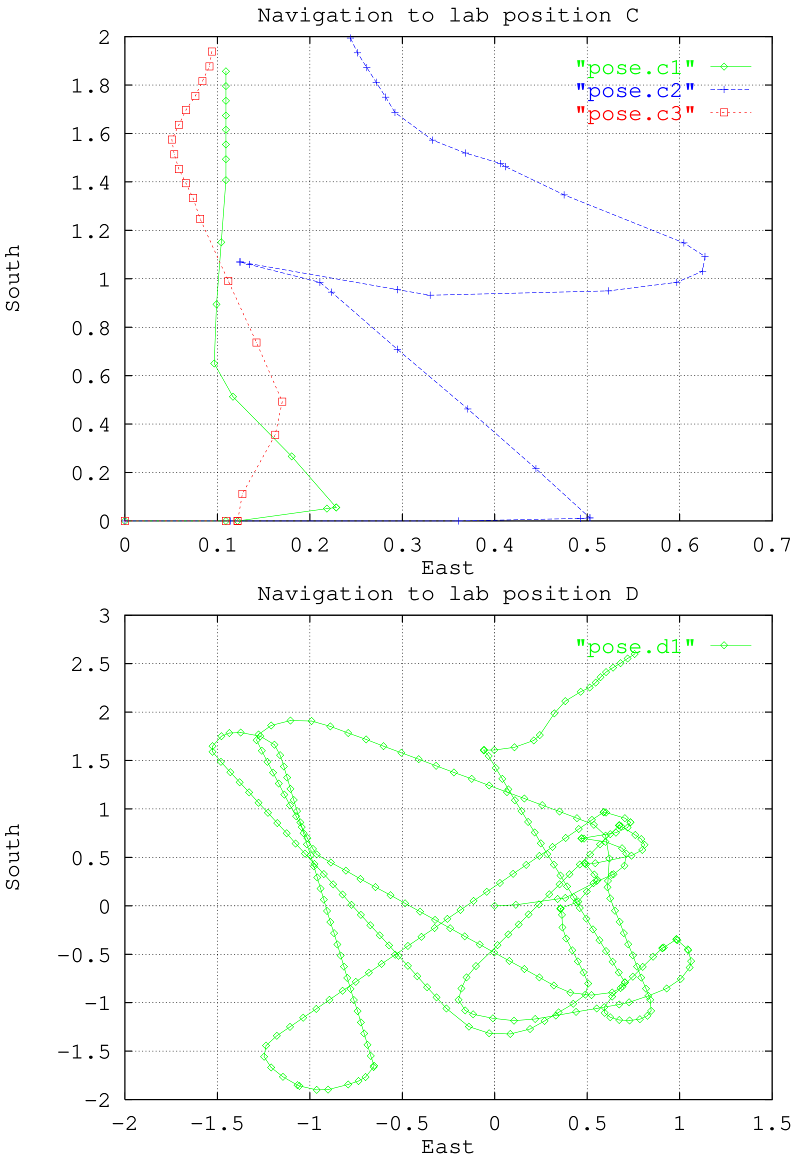

Figure 16. Results for learning with trash can in position C and D. The top figure shows three successful traces (starting at 0,0) of the robot navigating to lab position *C*. Note that in pose.c2 trace the robot temporarily loses the trash can but eventually gets back on track. The bottom figure shows a successful run when the trash can is in position *D*, which is initially unobservable to the robot.

generally less than 10). By contrast, in our work as well as in the ALVINN system, the input data has several hundred real-valued input variables, making it difficult to employ a top decision-tree type approach. The advantage, however, of using decision trees is that the learned knowledge can be easily converted into symbolic rules, a process that is much more difficult to do in the case of a neural net.

Symbolic learning methods have also been investigated for sensory concept learning. Klingspor et al. [12] describe a relational learning algorithm called GRDT, which infers a symbolic concept description (e.g. the concept *thru_door*) by generalizing user labeled training instances of a sequence of sensor values. A hypothesis space bias is specified by the user in the form of a grammar, which restrict possible generalizations. A strength of the GRDT algorithm is that it can learn hierarchical concept descriptions. However, a weakness of this approach is that it relies on using a logical description of the overall control strategy (as opposed to using a procedural reactive/declarative structure). Logical representations incur a computational cost in actual use, and their effectiveness in actual real-time robotics applications has not been encouraging.

7. Summary

This paper investigates how mobile robots can acquire useful sensory concepts from high-dimensional and noisy training data. The paper proposes two strategies for speeding up learning, based on decomposing the sensory input space, and learning multiple concepts simultaneously using a shared representation. The effectiveness of these strategies was studied in two tasks: learning feature detectors for probabilistic navigation and learning to recognize visual objects for recycling. A detailed experimental study was carried out using a Nomad 200 real robot testbed called PAVLOV. The results suggest that the strategies provide sufficient bias to make it feasible to learn high-dimensional concepts from limited training data.

Acknowledgements

This research was supported in part by an NSF CAREER award grant No. IRI-9501852. The authors wish to acknowledge the support of the University of South Florida Computer Science and Engineering department, where much of this research was conducted. We thank Lynn Ryan for her detailed comments on a draft of this paper.

Notes

1. PAVLOV is an acronym for Programmable Autonomous Vehicle for Learning Optimal Values.
2. In actuality, the state estimation procedure is more complex since we use an event-based *semi-Markov* model to represent temporally extended actions. However, for the purposes of this paper, we are simplifying the presentation.

3. Bias is generally defined as any criterion for selecting one generalization over another, other than strict consistency with the training set. It is easy to show that bias-free learning is impossible, and would amount to rote learning.
4. Although we do not discuss the details here, the robot employs a further processing phase to extract the rough geometrical aligment of the trash can opening in order to drop items inside it.

References

1. J. Borenstein. *Navigating Mobile Robots.* A.K. Peters, 1996.
2. R. Brooks. A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation*, 2(1), 1986.
3. R. Caruana. Multitask learning: A knowledge-based source of inductive bias. In *Proceedings of the Tenth International Conference on Machine Learning*, pages 41–48. Morgan Kaufmann, 1993.
4. T. Cassandra, L. Kaelbling, and J. Kurien. Acting under uncertainty: Discrete bayesian models for mobile-robot navigation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 963–972, 1996.
5. J. Connell and S. Mahadevan. *Robot Learning.* Kluwer Academic Publishers, 1993.
6. M. Dorigo. Introduction to the special issue on learning autonomous robots. In M. Dorigo, editor, *IEEE Transactions on SMC*, volume 26, pages 361–364. June 1996.
7. J. Engleberger. *Robotics in Service.* MIT Press, 1989.
8. S. Fahlman. Faster-learning variations on back-propagation: An empirical study. In *Proceedings of 1988 Connectionist Models Summer School.* Morgan Kaufmann, 1988.
9. J. Franklin, T. Mitchell, and Thrun. S., editors. *Robot Learning.* Number 2-3. Kluwer Academic Press, 1996. Special Issue of Machine Learning Journal, Vol 23.
10. A Jain. *Fundamentals of Digital Image Processing.* Prentice Hall, 1989.
11. N. Khaleeli. A robust robot navigation architecture using partially observable semi-markov decision processes. Master's thesis, University of South Florida, Tampa, FL, April 1997.
12. V. Klingspor, K. Morik, and A. Rieger. Learning concepts from sensor data. In J. Franklin, T. Mitchell, and Thrun. S., editors, *Recent Advances in Robot Learning.* Kluwer Academic Press, 1996. Special Issue of Machine Learning Journal, Vol 23.
13. S. Koenig and R. Simmons. Xavier: A robot navigation architecture based on partially observable markov decision process models. In D. Kortenkamp, P. Bonasso, and Murphy. R., editors, *AI-based Mobile Robots: Case-studies of Successful Robot Systems.* MIT Press, 1997.
14. A. Kosaka and A. Kak. Fast vision-guided mobile robot navigation using model-based reasoning and prediction of uncertainties. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2177–2186, 1992.
15. Salganicoff. M., L. Ungar, and R. Bajcsy. Active learning for vision-based grasping. In J. Franklin, T. Mitchell, and Thrun. S., editors, *Recent Advances in Robot Learning.* Kluwer Academic Press, 1996. Special Issue of Machine Learning Journal, Vol 23.
16. S. Mahadevan. Proceedings of the workshop on robot learning. Held in conjuction with the 11th International Conference on Machine Learning (IMLC '94). Online proceedings available at http://www.cs.cmu.edu/afs/cs.cmu.edu/user/mitchell/ftp/robot-learning.html, July 1994.
17. S. Mahadevan. Machine learning for robots: A comparison of different paradigms. In *Proceedings of the IROS '96 Workshop on Towards Real Autonomy*, 1996.
18. S. Mahadevan and N. Khaleeli. Mobile robot navigation using discrete-event markov decision process models. submitted.
19. J. McClelland and D.(editors) . Rumelhart. *Parallel Distributed Processing: Explorations in the Microstructure of Cognition.* Bradford Books, 1986.
20. T. Mitchell. *Machine Learning.* McGraw Hill, 1997.

21. I. Nourbakhsh, R. Powers, and S. Birchfield. Dervish: An office-navigating robot. *AI Magazine*, 16(2):53–60, 1995.
22. D. Pomerleau. Neural network based autonomous navigation. In C. Thorpe, editor, *Vision and Navigation: The CMU Navlab*. Kluwer Academic Publishers, 1990.
23. J. R. Quinlan. Induction of decision trees. *Machine Learning*, 1(1):81–106, 1986.
24. J. Shavlik and T. Dietterich. Inductive learning from preclassified training examples. In J. Shavlik and T. Dietterich, editors, *Readings in Machine Learning*. Morgan Kaufmann, 1990.
25. R. Sutton and A. Barto. *Reinforcement Learning: An Introduction*. MIT Press, 1998.
26. M. Tan. Cost-sensitive learning of classification knowledge and its application to robotics. *Machine Learning*, 13(1):7–33, 1993.
27. S. Thrun and T. Mitchell. Learning one more thing. Technical Report CMU-CS-94-184, Carnegie-Mellon University., 1994.