InfoMax Control for Acoustic Exploration of Objects by a Mobile Robot

Antons Rebguns[1,2], Daniel Ford[3], Ian Fasel[1]
[1]School of Information: Science, Technology, and Arts,
[2]Department of Computer Science,
[3]Department of Electrical and Computer Engineering
The University of Arizona
Tucson, AZ 85721-0077

Abstract

Recently, information gain has been proposed as a candidate intrinsic motivation for lifelong learning agents that may not always have a specific task. In the *InfoMax control* framework, reinforcement learning is used to find a control policy for a POMDP in which movement and sensing actions are selected to reduce Shannon entropy as quickly as possible. In this study, we implement InfoMax control on a robot which can move between objects and perform sound-producing manipulations on them. We formulate a novel latent variable mixture model for acoustic similarities and learn InfoMax polices that allow the robot to rapidly reduce uncertainty about the categories of the objects in a room. We find that InfoMax with our improved acoustic model leads to policies which lead to high classification accuracy. Interestingly, we also find that with an insufficient model, the InfoMax policy eventually learns to "bury its head in the sand" to avoid getting additional evidence that might increase uncertainty. We discuss the implications of this finding for InfoMax as a principle of intrinsic motivation in lifelong learning agents.

Introduction

Agents in dynamic environments, both biological and robotic, must continually deal with uncertainty about the environment and the objects they are interacting with. While most organisms (as well as statically placed sensor devices like security cameras) have the ability to passively sense their environment, one key feature of "intelligent" agents is the capacity for movement and manipulation, allowing them to actively sense the world and extract otherwise hidden information. Moving to an object and manipulating it is often the only way to get useful audio, tactile, or other sensations that provide information about shape, surface texture, deformability, and other material properties.

The problem of active selection of maximally informative actions has been treated extensively in statistics, where it is referred to as optimal experimental design, and in machine learning, where it is referred to as active learning (see Settles, 2009, for a review). These approaches typically do not address the unique concerns of organisms and robots embedded in physical space, in which the agent needs to traverse physical space, limited-range sensors need to be positioned and oriented properly, and certain manipulations must be done in a specific order.

Recently, *InfoMax control* has been proposed as a framework for addressing the needs of mobile agents with time-costs for moving or using sensors and information gathering actions. In the InfoMax approach, the information seeking problem is framed as a partially observable Markov decision process (POMDP), where the unknown states are the categories of the objects in the environment, and the actions at each time step are either to move (and if so, in what direction) or to select a sensing action to perform on the object at the current location. At each time step, the reward is the negative Shannon entropy of the unknown object categories, averaged over all objects in the room. The goal then is to find a policy which maps the current belief state to actions that will maximize the mutual information between the currently believed object categories and the resulting observations.

This paper extends InfoMax control to a more complex domain than has been studied in the past – namely, a mobile robot which learns to recognize a set of objects through the sounds produced when manipulating them (for instance, gasp, shake, drop, etc.). To do so, we develop an acoustic category model that provides the ability to take an arbitrary number of actions on an object to infer a *distribution* over acoustic category similarities. Our experiments with 10 objects (in worlds with 3 objects at different locations) show that learned InfoMax control policies do a good job of gathering information, resulting in high *post-hoc* classification accuracy. We also find the surprising result that, under an insufficiently complex object model, InfoMax policies will eventually stop taking actions in order not to accidentally gather information that might reduce confidence. In such cases, this behavior in fact does increase the expected intrinsic reward, however it leads to decreased classification accuracy. This leads us to speculate that while InfoMax is appealing as a natural, infant-like "curiosity" mechanism, it may be dangerous to rely on InfoMax if the robot's models may not be accurate and there is no way to automatically improve these models, or if there is prior knowledge about a robot's future tasks that could be better used to optimize the robot just for those tasks.

Background and Related Work

Inspired by results from developmental psychology (Watson and Fischer 1977), visual psychophysics (Najemnik and Geisler 2005) and single-cell recodings in monkeys (Bromberg-Martin and Hikosaka 2009), InfoMax has been used to model real-world behaviors, such as detection of social contingencies (Movellan 2005), control of eye saccades (Sprague and Ballard 2003; Butko and Movellan 2008; 2009; 2010), and head turns in socially interactive robots (Fasel et al. 2009). InfoMax-like ideas have also been used to learn the relationship between button presses and sounds (Sukhoy et al. 2010). Our current work is an extension of (Fasel et al. 2010), who used InfoMax in an "Information Foraging" agent which learned to take movement actions to reach objects scattered around the environment so that it could apply sensing actions on them to reduce uncertainty. That work was entirely in simulation, with very simple noise models on sensors which were fully independent. In this work we use a real robot which performs actions to extract acoustic properties from objects. In addition, our robot also must learn the dependencies between actions – for instance, that a *grasp* must precede any object manipulation, a *lift* must come before a *drop*, etc. – and when it is more informative in the long term to go to another object in the room rather than continue sensing the current closest object.

Our method for predicting categories from acoustics due to manipulations is based on (Sinapov, Wiemer, and Stoytchev 2009; Sinapov and Stoytchev 2010). In their work, a robot performed five different actions on an object and recorded the sound during the action. These sounds were converted to vectors of discrete symbols, which could then be compared to example sequences in a database using a general sequence alignment technique. The alignment distances from these comparisons were finally used for nearest-neighbors classification. In this paper, we describe a method for using these acoustic alignment distances to generate variable-width kernel-density estimates of sound "properties" in a probabilistic generative model, in which taking a particular action on an object is expected to produce a *mixture* of sounds properties. This probabilistic approach allows the robot to become increasingly certain of the object categories as it performs more actions, including repetitions of the same action, even if certain objects under certain actions tend to sound quite similar to several other objects under the same actions.

Robot, Objects, and Environment

We perform our experiments with a mobile manipulation robot, shown in Figure 1. The robot is constructed from a Videre "Erratic" mobile base, and is equipped with a wide variety of sensors for navigation, manipulation and object recognition. Two Hokuyo laser range finders on tilt servos are used for map building and dynamic obstacle avoidance. The robot has a stereo camera and a SwissRanger SR4000 for 3D depth estimation of objects. Audio is recorded through a USB condenser microphone mounted on front lip of the robot base. Manipulations are performed with a custom 7 degree-of-freedom arm, using Dynamixel

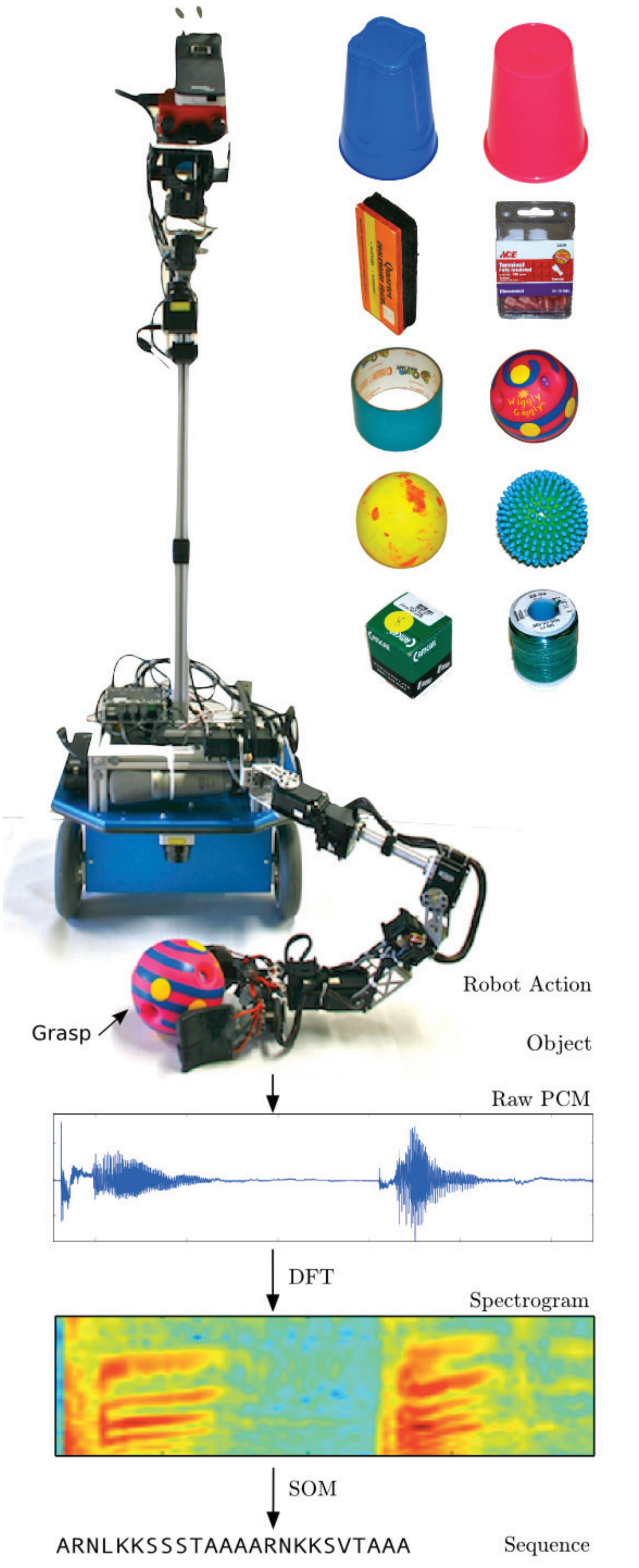

Figure 1: Top left*: The robot used in our experiments.* Top right*: The ten objects manipulated by the robot.* Bottom*: schematic of the process of taking an action on an object and then converting the acoustic signal into a sequence of tokens.*

(Bioloid) servos, with 8 force sensitive resistors in the claw used as a crude sense of touch. The robot can operate autonomously on batteries, and while the robot has an 802.11n wireless connection, all processing is performed onboard using a quad-core Intel Core i7 laptop. We use the Willow Garage Robot Operating system (ROS, Quigley et al., 2009) for coordination of different modules, such as localization, path planning, and acoustic sensing. Planning for arm movements and grasps based on the 3D localization of objects are used using OpenRAVE (Diankov 2010) and the Point Cloud Library (Rusu and Cousins 2011). Complete specifications for the robot and arm, including drivers for all the hardware and all source code for the experiments in this paper are freely available.[1]

For all experiments in this paper, we use ten objects, pictured in Figure 1. At the beginning of each episode, some number (typically 3-5) of the objects are placed around the robot in a ring. Time is divided into decision points[2], and at each decision point the robot can move to the closest object on the left or right, or it can perform one of six manipulations on each object: *push, lift, shake-pitch, shake-roll, set-down, drop*, where *shake-pitch* shakes the object up-and-down, and *shake-roll* shakes the object by rotating its wrist from side-to-side. Sound is recorded during and immediately following each of these actions, and the robot uses this information to update its belief about the object category (as described below). At each timestep, given its updated beliefs about the object category, the robot selects and performs a new action, until it has reached a maximum time-limit.

Acoustic similarity

When an object is acted upon, a sound may be produced depending on the material properties of the object, the action dynamics, and the properties of other interacting objects or surfaces. For actions such as pushing or shaking, the sound may continue for as long as the action occurs. For actions such as dropping an object, the sound happens immediately *after* the action is performed, and may be very short.

To compare different sounds of different lengths, we used the method described in (Sinapov, Wiemer, and Stoytchev 2009). In this method, a database of 20-100 sounds (captured at 44.1kHz) for each object-action pair is first created by the robot. Then for each action, all the sound sequences for that action are transformed into a timeseries of 17 frequency bins using a fast Fourier transform (FFT) with overlapping 11.6ms windows. The FFTs are then used to train a self organizing map (SOM). Using this representation, any acoustic signal can be transformed into a sequence indicating which SOM nodes are most highly activated in each time window. Two audio sequences can then be compared by finding an optimal global string alignment using the Needleman-Wunsch algorithm, which returns an alignment disance. These pairwise distances are finally used to estimate class probabilities $p = (p_1, ..., p_N)$ for N classes using a k-nearest neighbors (kNN) estimation technique. Specifically, let $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{M}$ be a database of M examples, where x_i is the ith audio sequence and y_i is the category label. Then for an example audio sequence x, the probability of class c is estimated by

$$p_c = \Big(\sum_{y \in B_k(x,\mathcal{D})} \delta(i, y) + \epsilon \Big) \;/\; Z(x) \qquad (1)$$

where $B_k(x, \mathcal{D})$ are the k nearest neighbors to x in the database $\mathcal{D}$ using the Needleman-Wunsch alignment distance, δ is the Kronecker delta function (i.e., $\delta(u, v) = 1$ if u, v are equal and zero otherwise), ϵ is a small regularization term (in our case 0.01), and $Z(x)$ is the partition function to ensure the probabilities sum to one.

Multi-action acoustic category model

The method above gives us a probability estimate for a single sound produced by taking one action on an object. In order to combine the result of multiple actions (including repetitions of the same action) we developed two candidate models. The first model maps sound similarities directly to object category probabilities. The second uses an intermediate representation where object categories represent a *distribution* of sound similarities given each action.

For both models, we note that if we made the assumption that each observation is conditionally independent given the action and object, then the class probabilities conditioned on all observed sounds could be computed simply by taking the product of the above probabilities and normalizing. This assumption is too simplistic however. For instance, if an action is repeated on an object, then taking products of the kNN probability estimates would usually result in an overly high confidence for one category even if that category only gets slightly higher probability on each individual trial. Therefore, both of our two candidate models try to deal with the problem of lack of independence.

Model 1: Our first model attempts to handle this issue by modifying the probability estimate in eq. (1) so that, given T acoustic measurements of an object,

$$p_c = \Big(\sum_{t=1}^{T} \sum_{y \in B_k(x_t,\mathcal{D})} \delta(c, y) \Big) \;/\; Z(x) \qquad (2)$$

This modification has the result that multiple repetitions of a particular action leads to less over-confident probability estimates than if the sounds produced by the actions were assumed conditionally independent. To combine across actions, the per-action probabilities are averaged. Each per-action probability is initialized to uniform and is replaced with the result of eq. (2) once the action has been taken.

Model 2: Our second model takes a more principled approach to modeling the fact that an action on an object legitimately yields a distribution over object similarities. Our reasoning is that because the underlying causes for sounds are a complex relationship between shape and material properties of the object, gripper, and floor, acoustic similarities are only indirectly related to the object category through these (always hidden) properties. Therefore it is important to explicitly model the fact that some objects sound somewhat like other objects under certain actions (for instance, most empty containers sound quite similar when shaken).

[1] `http://code.google.com/p/ua-ros-pkg`

[2] Technically this makes this a partially observable *semi*-MDP.

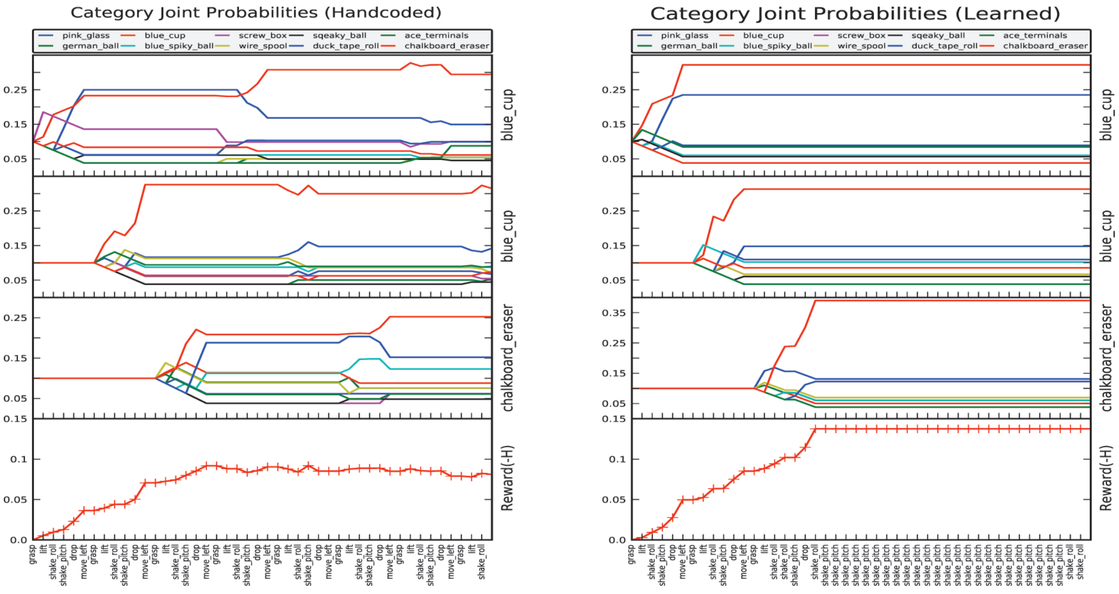

Figure 2: *Example beliefs and rewards for three objects as a function of time, using acoustic Model 1.* Left: *the heuristic (i.e., Hand-coded) policy,* Right*: the learned InfoMax policy. The top three graphs in each sugfigure shows the evolution of the robot's beliefs about the object categories as it takes actions. The bottom shows the scaled negative entropy reward over time. Note that after visiting each object once, the reward trends downward for the heuristic policy, whereas the learned policy has learned to take illegal actions once it has visited each object, thus keeping its reward high (best viewed in color).*

We address this by using a generative model in which each action-object category specifies a Dirichlet distribution from which a particular distribution of sound-similarities (estimated using eq. (1)) are sampled. Thus the probability of generating probabilities ϕ by taking action a on an object of category i is

$$p(\phi|a,i) = \frac{\Gamma(\sum_{j=1}^{N} \alpha_{aij})}{\prod_j \Gamma(\alpha_{aij})} \prod_{k=1}^{N} \phi_k^{\alpha_{aik}-1} \qquad (3)$$

where $\alpha_{ai} = (\alpha_{ai1}, ..., \alpha_{aiN})$ are the parameters for a Dirichlet distribution over acoustic probabilities for object category i under action a. This model treats elements of ϕ as functionally different than category labels – they could be replaced with a different type of sound feature probability measure using some other method (for instance a Gaussian mixture model over spectral features). The posterior probability of a category given a *set* of actions can now be calculated by taking the product of the probabilities estimated with eq. (3) for each action and then normalizing.

Learning an InfoMax controller

Once we have specified an acoustic model, we can use reinforcement learning to find a policy for selecting actions. Let q_t be a d-dimensional vector combining the robot's current beliefs about the objects and its known internal state (described below), and define the set of possible actions $A = \{$*push, lift, shake-pitch, shake-roll, set-down, drop, move-left*$\}$. Then let the function $F_\theta : Q \rightarrow A$ be a deterministic controller with k-dimensional parameter θ which at each time t takes as input a state-variable q_t and outputs an action a_t.

Representation

To construct q_t, let p' be an egocentric representation of the agent's current beliefs, i.e., p'_1 is the agent's beliefs about the object directly in front of it and p'_2 through p'_M are the agent's beliefs about the remainder of the M objects, arranged from left to right. Then let $q_t = (p'_1, ..., p'_M, c'_1, ..., c'_M, \psi(t))$ where c' is an egocentrically arranged vector of counters of how often each action has been taken on each object, and $\psi(t) = (\psi_1(t), \psi_2(t), \psi_3(t))$ is a vector of radial basis functions of the time t, which allows the learned policy to depend on the number of steps taken in an episode.

Let an episode (or history) $h = (q_1, a_1, ..., q_T, a_T)$ be a sequence of T state-action pairs induced by using a controller with parameters θ. We can then define the reward at time t of episode h as the (scaled) negative Shannon entropy of the belief distribution, averaged over all objects, i.e.,

$$\mathcal{R}(q_t|h) = \frac{1}{a}\Big(\sum_{i,k} p_{ik}^{(t)} \log p_{ik}^{(t)} + b\Big) \qquad (4)$$

where $p_{ik}^{(t)}$ is the agent's belief that object in position k is an instance of category i based on the experiences in h up to time t. Constants a and b are simply to scale reward to $[0, 1]$ and are fixed beforehand.

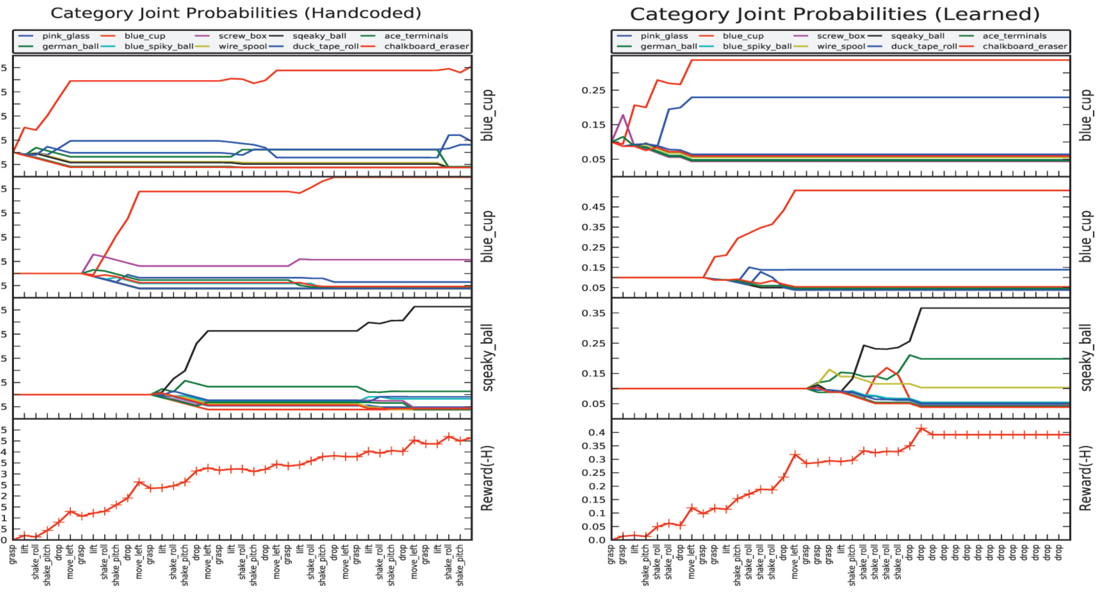

Figure 3: *Example beliefs and rewards for three objects as a function of time, using acoustic Model 2.* Left: *the heuristic (i.e., Hand-coded) policy,* Right*: the learned InfoMax policy. Note that in this case uncertainty continues to decrease as repeated actions are taken. However the learned policy, though better, still stops gathering information after awhile.*

Policy Learning

In the current setting, the goal of the learning algorithm is to find parameters θ that maximizes the expected total reward over episodes of fixed length L, i.e., maximize the objective

$$\Phi(\theta) = E_h[\sum_{t=1}^{L} \mathcal{R}(q_t|h)p(h|\theta)]. \quad (5)$$

Although many optimization algorithms could work in this situation, in this paper we learn the parameters from experience using the *Policy Gradients with Parameter Exploration* (PGPE) algorithm (Sehnke et al. 2010), a model-free reinforcement learning algorithm for POMDPs which performs exploration by sampling in the parameter space of a controller. Rather than computing gradients for the objective function with respect to the controller's parameters, a gradient is instead estimated over a set of *hyper*paramters from which parameters θ of a controller are sampled. For completeness we give a brief description here but we defer to (Sehnke et al. 2010) for details.

Let θ_n be a d-dimensional vector, so that we can rewrite $\theta = (\theta_1, ..., \theta_k)$ as a set of weight vectors for the function: $F_\theta(q_t) = argmax_a\, \theta_a q_t$, i.e., it calculates one linear combination of the inputs q_t per action, then selects the maximum scoring action. For each learning episode, each parameter of θ is independently sampled from a one dimensional normal distribution with mean and variance μ_i, σ_i, which we collect together as $\rho = (\mu_1, ...\mu_d, \sigma_1, ...\sigma_d)$.

PGPE performs a gradient descent procedure over ρ to optimize policies of the form

$$p(a_t|q_t, \rho) = \int_\theta p(\theta|\rho)\delta(F_\theta(q_t), a_t)\, d\theta \quad (6)$$

Let $r(h) = \mathcal{R}(q_T|h)$, and let H be the set of all possible histories. The expected reward is then given by

$$J(\rho) = \int_\Theta \int_H p(h, \theta|\rho)r(h)\, dh\, d\theta \quad (7)$$

Differentiating with respect to ρ and using the identity $\nabla_x y(x) = y(x)\nabla_x \log y(x)$, we have

$$\nabla_\rho J(\rho) = \int_\Theta \int_H p(h, \theta|\rho)\nabla_\rho \log p(h, \theta|\rho)r(h)\, dh\, d\theta \quad (8)$$

Noting that h is conditionally independent of ρ given θ, this can be estimated with a *sample* of histories, by repeatedly choosing θ from $p(\theta|\rho)$ and then running the agent with this policy to generate a history h. Thus, given a set of rollouts $(h^{(1)}, ..., h^{(N)})$ generated from sample controllers with parameters $(\theta^{(1)}, ..., \theta^{(N)})$,

$$\nabla_\rho J(\rho) \approx \frac{1}{N}\sum_{i=1}^{N} \nabla_\rho \log p(\theta^{(i)}|\rho)r(h^{(i)}) \quad (9)$$

Stochastic gradient descent can now be performed until a local maximum is found. (Sehnke et al. 2010) show that with proper bookkeeping, each gradient update can be efficiently performed using just two symmetric samples from the current ρ.

Experiments

We performed a number of experiments to compare policies using the two different acoustic recognition models. As described in the previous sections, an initial database of 20-200 samples per action-object pair was first collected.

These samples were used to train the SOMs used to convert acoustic samples into strings. Using the string alignment distances, we computed the kNN results for each sample with that sample left out in a leave-one-out procedure. For model 2, these kNN results were used to estimate the parameters of the action-object specific Dirichlet distributions. Given this database, we could then perform experiments in simulation by sampling sound-action sequences called for by the controller.

For each model, we performed experiments in two conditions. In the *independent actions* condition, any action could be taken at any time, and the beliefs were updated as described above at each timestep. In the *dependent actions* condition, we enforced the sequential dependencies inherent in manipulation, e.g., for most actions the object must first be in-hand due to a previous *grasp*, a *drop* or *shake* must have been preceeded by a *lift*, a *drop* makes the object no longer in-hand, etc. In this case, if the robot attempted an illegal action, the beliefs were simply not modified, similar to a *move* action, and otherwise everything remained the same.

We created a large number of conditions, varying the number of objects, the size of the horizon, and variations in the acoustic models. For each setting, we trained InfoMax policies for 10,000 episodes of PGPE. For each episode, a set of object categories are randomly chosen, and the robot's beliefs about the object categories are initialized to uniform. Then at each timestep, provided the composite state vector q_t, an action is selected using the current policy. The robot's joint beliefs are updated if appropriate, and the intrinsic reward is returned. Each full learning trial of PGPE was repeated 16 times and all results show either averages across these 16 experiments or example rollouts from the θ resulting in the highest average reward across all runs.

Results

Most of our results can be summarized by studying the case of 3-objects with 45 step episodes in a few different conditions. All graphs show the *dependent actions* case. In most cases we compare the performance of a learned policy to the performance of the heuristic "hand-coded" policy.

Our first result is that, in all cases, InfoMax learned a good policy that tended to perform all actions on each object and then move. As shown in Tables 1 and 2, the learned policies always led to accurate predictions about all three objects by the end of the episode, and were competitive with heuristic policies. When the dependencies between manipulations were enforced, InfoMax also always learned policies that chose legal orderings of actions.

However as we can see in Figures 2 and 3, the learned policies often had unintuitive characteristics. Each graph shows an example rollout of a policy. The horizontal axis shows the action performed at timestep t. The vertical axis of the top three subplots shows the category probabilities, and the bottom subplot shows the intrinsic reward after executing that step in the policy. On the left of each figure, we can see that the hand coded policy executes a fixed *grasp, lift, shake-roll, shake-pitch, drop, move-left* sequence repeatedly. The right shows the learned policy, which was adaptive.

Fig. 2 shows the problem for acoustic Model 1: after performing each action on each object once, the reward ("certainty") tends to decrease. This is because additional evidence leads to less peaked distributions over the sound-similarities. However under Model 2 (Fig. 3), this is not the case – more evidence about the *distribution* of sound similarities continues to reduce entropy in the object category beliefs. In Fig. 4 we can these trends clearly from the reward-per-step averaged across 100 trials for each policy.

Fig. 2 shows that the InfoMax policy under Model 1 has learned a clever trick to avoid increasing uncertainty: After taking each action on each object exactly once, it takes illegal actions that don't change the beliefs, so that it can "bury its head in the sand" from that point on. Fig. 3 shows that Model 2 improves this situation somewhat – indeed the policy does repeat particularly informative *shake* actions on each object – however it too starts taking illegal actions after visiting each object. However in this case, the policy still achieves superior classification accuracy.

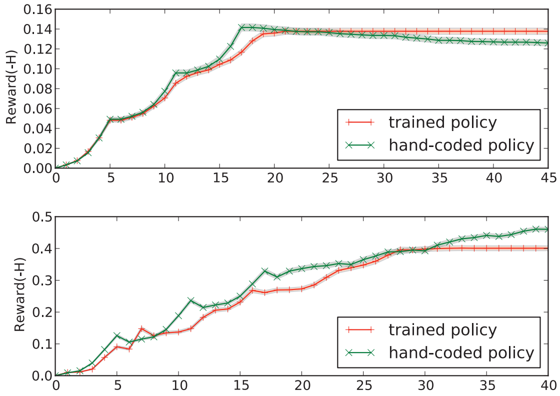

Figure 4: *Average reward per step for Models 1 (*top*) and 2 (*bottom*).*

Finally, we can see in Tables 1 and 2 that with Model 1, the handcoded policy ultimately outperforms the learned policy even though its uncertainty is greater on average. However using acoustic Model 2, the learned policy, which repeats some actions on objects, is consistently more accurate than the hand-coded policy. This makes sense because under this model the agent can reduce uncertainty by better estimating a *distribution* over sound similarities instead of seeking a single, maximum sound similarity.

Step Number	1	4	9	13	18	30	45
Model 1 Learned	**22.7**	**36.0**	**60.7**	72.7	94.0	94.7	94.7
Model 1 Handcoded	20.0	34.3	59.0	**75.0**	**94.7**	**98.0**	**99.3**
Model 2 Learned	22.0	35.0	47.0	67.0	77.3	**98.0**	98.0
Model 2 Handcoded	19.3	35.7	59.0	73.7	93.7	97.3	**99.3**

Table 1: *Classification accuracy (percent) per step with dependent actions*

Step Number	1	4	9	13	18	30	45
Model 1 Learned	29.7	38.3	**64.0**	69.7	**94.0**	99.0	98.7
Model 1 Handcoded	20.0	36.7	53.3	69.7	81.3	98.3	**99.3**
Model 2 Learned	**37.0**	**40.7**	52.0	**71.0**	75.0	**99.3**	**99.3**
Model 2 Handcoded	20.3	37.7	49.3	68.0	82.3	97.7	97.7

Table 2: *Classification accuracy (percent) per step with independent actions*

Discussion and Conclusions

From these results we can draw a few conclusions. First, we have shown that InfoMax control policies can indeed be learned with complex, real-world robot sensors and manipulation actions. We also have shown that our hierarchical acoustic model consistently improves accuracy by modeling distributions of sound similarities. Combining this improved model with InfoMax control, our learning agent is consistently better at identifying object categories than an agent that uses a non-adaptive hand-coded policies with either acoustic model.

We have also found the unintuitive result that the negative entropy reward can lead to policies that deliberately avoid new evidence in order not to increase uncertainty, which results in decreased accuracy. This gives us a sense of two different emergent "personalities": one which seeks clear-cut distinctions, and another which seeks more complete knowledge about the world. This raises questions about when InfoMax control makes sense. In a lifelong learning or developmental robotics setting, where the robot can't always get ground-truth labels from a human, InfoMax might be a good reward to use until a specific task is provided. However this could lead to pathological behaviors if the internal models are not sufficient to capture important dependencies in the world, and it has no way of improving the model itself. This suggests that if the robot builder knows beforehand how the robot will be used, for instance to classify or fetch objects, then it may be better to directly optimize the policy for that goal in order to compensate for possible deficiencies in the underlying models.

Acknowledgements

This work was funded by ONR contract N00014-09-1-0658 and DARPA contract N10AP20008.

References

Bromberg-Martin, E., and Hikosaka, O. 2009. Midbrain dopamine neurons signal preference for advance information about upcoming rewards. *Neuron* 63(1):119–126.

Butko, N., and Movellan, J. 2008. I-POMDP: An infomax model of eye movement. In *7th IEEE International Conference on Development and Learning, 2008. ICDL 2008*, 139 –144.

Butko, N. J., and Movellan, J. R. 2009. Optimal scanning for faster object detection. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Butko, N. J., and Movellan, J. R. 2010. Infomax control of eye movements. *IEEE Transactions on Autonomous Mental Development* 2(2):91–107.

Diankov, R. 2010. *Automated Construction of Robotic Manipulation Programs*. Ph.D. Dissertation, Carnegie Mellon University, Robotics Institute.

Fasel, I.; Ruvolo, P.; Wu, T.; and Movellan, J. 2009. Infomax control for social robots. In *NIPS Workshop on Probabilistic Approaches for Robotics and Control*.

Fasel, I.; Wilt, A.; Mafi, N.; and Morrison, C. 2010. Intrinsically Motivated Information Foraging. In *Proceedings of the 9th IEEE International Conference on Development and Learning (ICDL 2010)*, 604–609.

Movellan, J. 2005. *An Infomax Controller for Real Time Detection of Social Contingency*. In *Proceedings of the 9th IEEE International Conference on Development and Learning (ICDL 2005)*.

Najemnik, J., and Geisler, W. 2005. Optimal eye movement strategies in visual search. *Nature* 434:387–391.

Quigley, M.; Conley, K.; Gerkey, B. P.; Faust, J.; Foote, T.; Leibs, J.; Wheeler, R.; and Ng, A. Y. 2009. ROS: an open-source robot operating system. In *ICRA Workshop on Open Source Software*.

Rusu, R. B., and Cousins, S. 2011. 3D is here: Point Cloud Library (PCL). In *IEEE International Conference on Robotics and Automation (ICRA)*.

Sehnke, F.; Osendorfer, C.; Rucksties, T.; Graves, A.; Peters, J.; and Schmidhuber, J. 2010. Parameter-exploring policy gradients. *Neural Networks* 551–559.

Settles, B. 2009. Active learning literature survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison.

Sinapov, J., and Stoytchev, A. 2010. The odd one out task: Toward an intelligence test for robots. In *Proceedings of the 9th IEEE International Conference on Development and Learning (ICDL 2010)*, 126–131.

Sinapov, J.; Wiemer, M.; and Stoytchev, A. 2009. Interactive learning of the acoustic properties of household objects. In *IEEE International Conference on Robotics and Automation, 2009. (ICRA'09)*, 2518–2524. IEEE.

Sprague, N., and Ballard, D. 2003. Eye movements for reward maximization. In *In Advances in Neural Information Processing Systems 15*. MIT Press.

Sukhoy, V.; Sinapov, J.; Wu, L.; and Stoytchev, A. 2010. Learning to Press Doorbell Buttons. In *Proceedings of the 9th IEEE International Conference on Development and Learning (ICDL)*, 132–139.

Watson, M. W., and Fischer, K. W. 1977. A Developmental Sequence of Agent Use in Late Infancy. *Child Development* 48(3):828–836.