Imitation and Reinforcement Learning with Heterogeneous Actions

Bob Price
Department of Computer Science
University of British Columbia
Vancouver, B.C., Canada V6T 1Z4
price@cs.ubc.ca

Craig Boutilier
Department of Computer Science
University of Toronto
Toronto, ON, Canada M5S 3H5
cebly@cs.toronto.edu

Abstract

We study the problem of accelerating reinforcement learning through the observation and *implicit imitation* of expert agents (mentors) acting in the same domain. In this paper, we consider problems that arise when the learner and mentor have heterogeneous actions. We extend an earlier implicit imitation model to allow for feasibility testing (determining whether a specific mentor action can be duplicated) and repair (discovering a "plan" that simulates a mentor's trajectory) and demonstrate empirically that both of these components allow learning agents to learn much more readily than standard RL agents and implicit imitation agents without these extended capabilities.

1 Introduction

Cooperative multiagent systems rely on shared models and communication to coordinate their actions in a common environment. While many researchers have examined explicit communication systems, we have argued (as have others) that implicit communication techniques such as imitation increase the range of applications for multi-agent systems and pose interesting cognitive models of interaction in agent societies [4, 14]. In an imitation model with implicit communication, agents can learn from others: without communicating an explicit context for the applicability of a behaviour [2]; without the need for a pre-existing communication protocol; in competitive situations where agents are unwilling to share information; and even when the other agents are unwilling to fulfil a teacher role. The ability of imitation to effect transfer between agents has been demonstrated by a number of researchers for a range of domains [6, 8, 1, 3, 9, 17, 11] These domains, however, have primarily dealt with agents imitating other agents with essentially the same action set as themselves. One of the major aspirations of imitation research is the promise of learning from agents which are "different" in some way from the learner.

In previous work [14] we developed a model for implicit imitation in reinforcement learning (RL) in which an agent could learn how to act more effectively by observing the state transitions of mentor agents and using these to influence its estimates of it value function. Though we made no assumption that the learner shared the same objectives as the mentors, we did rely crucially on the fact that actions were *homogeneous*: every action taken by a mentor corresponded to some action available to the learner. In this paper, we relax this assumption and introduce several mechanisms that allow acceleration of RL in presense of *heterogeneous actions*. Specifically, we introduce two notions: *action feasibility testing*, which allows the learner to determine whether a specific mentor action can be duplicated; and *k-step repair*, in which a learner attempts to determine whether it can approximate the mentor's trajectory. Both of these concepts are used to modify the influence that mentor observations have on the learner's estimate of its own value function.

Our work can be viewed (loosely) as falling within the formal imitation framework proposed by Nehaniv and Dautenhahn [13], who propose viewing imitation as the construction of mappings between the states, actions, and goals of different agents (see also the abstraction model of Kuniyoshi at al. [8]). However, key differences include the fact that we assume that state-space mappings are given, that the mentor's actions are not directly observable, that the objectives (goals) of the mentor and learner may be different, and that our environments are stochastic. Furthermore, we do not require that the learner explicitly try to duplicate the behavior of the mentor. In this way, our model differs from "following" and "demonstration" models often used in robotics [1, 9, 6]. However, the repair strategies we invoke do bear some relation to "following" models.

2 Implicit Imitation with Homogeneous Actions

We begin with a brief review of our basic framework and the implicit imitation model of [14]. We assume two agents, a

mentor m and an *observer* o, acting in a fixed environment.[1] The observer is a model-based reinforcement learner that can observe aspects of the mentor's behavior. Specifically, we assume the observer (or learner) is learning to control an MDP with states S, action A_o and reward function R_o. We use $\Pr_o(t|s,a)$ to denote the probability of transition from state s to t when action a is taken. The mentor too is controlling an MDP with the same underlying state space (we use A_m, R_m and $\Pr_m$ to denote this MDP).

We make two assumptions: the mentor is an "expert" and is thus implementing a stationary policy π_m, which induces a Markov chain $\Pr_m(t|s) = \Pr_m(t|s,\pi_m(s))$ over S.; and for each action $\pi_m(s)$ taken by the mentor, there exists an action $a \in A_o$ such that the distributions $\Pr_m(\cdot|s,\pi_m(s))$ and $\Pr_o(\cdot|s,a)$ are the same. This latter assumption is the *homogeneous action assumption* and implies that the learner can duplicate the mentor's policy. We do not assume that the learner knows *a priori* the identity of this action a (for any given state s), nor that the learner *wants* to duplicate this policy (the agents may have different reward functions). Since the learner can observe the mentor's transitions (though not its actions directly), it can form estimates of the mentor's Markov chain, along with estimates of its own MDP (transition probabilities and reward function).

We define the *augmented Bellman equation* as follows:

$$V(s) = R_o(s) + \gamma \max\left\{ \max_{a \in A_o}\left\{ \sum_{t \in S} \Pr_o(t|s,a) V(t) \right\}, \sum_{t \in S} \Pr_m(t|s) V(t) \right\} \quad (1)$$

This is the usual Bellman equation with an extra term added, the second summation, denoting the expected value of duplicating the mentor's action $\pi_m(s)$. Since this (unknown) action is identical to one of the observer's actions, the term is redundant and the augmented value equation is valid. Furthermore, under certain (standard) assumptions, we can show that the estimates of the model quantities will converge to their true values; and an *implicit imitation learner* acting in accordance with these value estimates will converge optimally under standard RL assumptions.[2] More interesting is the fact that by acting in accordance with value estimates produced by augmented Bellman backups, an observer generally converges much more quickly than a learner not using the guidance of a mentor. As demonstrated in [14], implicit imitators typically accumulate reward at a higher rate earlier than standard (model-based) RL-agents, even when the mentor's reward function is not identical to the observer's.

We note that the influence of a mentor can be misleading at states the mentor visits infrequently. Even when adopting a deterministic policy, action noise can cause the mentor to visit certain states very infrequently, leading to very inaccurate estimates of the mentor's Markov chain at such states (compared to the learner's own estimated action models). In such cases, we would like to suppress the mentor's influence: we do this by using model confidence in augmented backups. For the mentor's Markov chain and the observer's action transitions, we assume a Dirichlet prior over the parameters of each of these multinomial distributions. From sample counts of mentor and observer transitions, the learner updates these distributions. Using a technique inspired by Kaelbling's interval estimation method [7], we use the variance in our estimated (Dirichlet) distributions for the model parameters and use these to construct lower bounds on both the augmented value function encorporating the mentor model and an unaugmented value function based stricly on the observer's own experience. If the lower bound on the augmented value function is less than the lower bound on the unaugmented value function, we suppress the influence of the mentor and use an unaugmented Bellman backup.

3 Implicit Imitation with Heterogeneous Actions

When the homogenity assumption is violated, the implicit imitation framework described above can cause the learner to perform very poorly. In particular, if the learner is unable to make the same state transition (or a transition with the same probability) as the mentor at a given state, it may drastically overestimate the value of that state. Furthermore, there is no mechanism for removing the influence of the mentor's Markov chain on value estimates—the observer can be extremely (and correctly) confident in the mentor's model. The problem lies in the fact that the augmented Bellman backup is justified by the assumption that the observer can duplicate *every* mentor action.

To overcome this difficulty, we propose two techniques that allow observers to retain the guidance of mentors, but suppress the guidance when it is apparent that it is misleading. The more fundamental of these, but in some sense the more straightforward, is action feasibility testing: intuitively, when the learner is sure that it cannot duplicate the mentor's action at a given state, it suppresses the effect of augmented backups at that state (reverting to standard Bellman backups).[3] The technique is simple and eliminates the "lockup" effect sometimes observed in the basic implicit imitation framework when agents are permitted heterogeneous actions. Unfortunately, this can sometimes cause useful guidance (in the form of higher value estimates) to be "cut off" in certain cases where that guidance would be useful. Specifically, when the learner can "repair"

[1] The extension to multiple mentors is straightforward [14].

[2] We assume that an appropriate exploration strategy is being used and that it is influenced by estimated value; i.e., the learner is more likely to choose actions with higher estimated value.

[3] The decision is binary; but we could envision a smoother decision criterion that measures the extent to which the mentor's action can be duplicated. We do not pursue this generalization here.

the mentor's trajectory by finding a (short) sequence of its own actions that lead to the same state as the infeasible action, the value guidance is likely appropriate. For this reason, we introduce the notion of *k-step repair* and a method for deciding when to allow mentor guidance to persist at a state despite the infeasibility of the mentor's action for the observer.

3.1 Action Feasibility Testing

The Dirichlet distributions used by our model-based RL-agent, can be used to find the variance associated with a transition probability estimate and this estimate can be used to test the feasibility of a mentor's action. To examine a simple case, suppose that there are only two successor states, t and u, for a specific action a_o taken at s (thus we estimate only one probability $\Pr_o(t|s,a_o)$). Now we imagine that the mentor's action is similarly restricted and the mentor's Markov chain at that state is modeled by $\Pr_m(t|s)$. We could test statistically whether the two actions a_o and the mentor's action are the same by performing a difference of means test using the hypothesis that the mean probability of getting to state t is the same for both actions. Under this hypothesis we use the pooled variance of the two statistics which is computed by weighting the variances according to the number of samples used for each statistic.

$$\frac{Pr(t|a_1) - Pr(t|a_2)}{\sqrt{\frac{n_1(t|a_1)Var(t|a_1)+n_2(t|a_2)Var(t|a_2)}{n_1(t|a_1)+n_2(t|a_2)}}} > Z_{\alpha/2} \qquad (2)$$

The Dirichlet distribution is highly non-normal for small sample size, so we construct our test criterion $Z_{\alpha/2}$ using the Tchebycheff inequality which is valid for any distribution. When the value of the left side of Equation 2 is greater than the right, we conclude that the actions are different and that there is no point in having the observer attempt to duplicate the mentor.

Generally, however, we will have a number of possible outcomes for an action (not just two) so we must perform a multivariate difference of means test. For well-behaved distributions (e.g., normal) there exist multi-variate difference of means tests [15]. The work specific to multivariate testing of Dirichlet or generalized beta distributions assumes a sufficient number of samples to make the bounds computed reasonably tight [5]. A second method applicable to multivariate Dirichlet distributions is the Bonferroni Test [16] which allows one to construct a multivariate test from univariate components. It makes no assumptions about normality or independence and in comparison with techniques like [5], it has been shown to give good results in practice [10]. Since it is also easy to implement and fast to compute, we employed the Bonferroni method in our implementation.

The idea behind the Bonferroni test is to perform a multivariate hypothesis test by conjoining several single variable tests. More generally, we might have a set of r specific

```
FUNCTION feasible(m,s) : Boolean
    FOR each a_i in A_o do
        allSuccessorProbsSimilar = true
        FOR each t in successors(s) do
            μ_Δ = Pr_o(t|s,a) − Pr_m(t|s)
            z_Δ = μ_Δ √(var_o(t|s,a) + var_m(t|s))
            IF z_Δ > z_{α/r}
                allSuccessorProbsSimilar = false
        IF allSuccessorProbsSimilar
            return true
    return false
```

Figure 1: Action Feasibility Testing

hypotheses $E_1, E_2, \ldots, E_r$ that we wish to test simultaneously. Let $\bar{E}_i$ be the complementary hypothesis of E_i. The Bonferroni inequality tells us:

$$Pr\left[\bigcap_{i=1}^{r} E_i\right] \geq 1 - \sum_{i=1}^{r} Pr\left[\bar{E}_i\right]$$

Thus we can obtain a probability of α for the joint hypothesis $\bigcap_{i=1}^{r} E_i$ by testing each of the r complementary hypotheses $\bar{E}_i$ at α/r. The individual hypotheses E_i do not have to be independent. In testing for action equivalence, our individual hypotheses correspond to tests to see if the transition probability to a particular successor state is the same for both actions and the joint hypothesis is that all successor state transition probabilities are the same for both actions. We therefore set r to be the number of successor states.

To summarize, we test the distribution of successor states for the mentor's unknown action against the distribution of successor states for each of the observer's actions using a Bonferroni test. If all of the observer's experience-based action models are rejected, then it concludes that the mentor's action is infeasible and the influence of the model derived from mentor observations is suppressed. The algorithm is summarized in Figure 1.

4 k-Step Repair

Even if an observer cannot duplicate a mentor's primitive action at a particular state, guidance from the mentor may still be useful if the trajectory of the mentor through the state space is broadly "similar" to a feasible trajectory for the observer. We can capture this notion of "similarity" by augmenting feasibility testing with a device that encourages the learner to find these "similar" trajectories.

For example, suppose the observer is at state s and the mentor has been observed to make the transition from state s to state t to state u enough times that the observer's estimates of $\Pr_m(t|s)$ and $\Pr_m(u|t)$ are very confident (see Figure 2). Suppose also, that state u is a highly rewarding state for both the mentor and observer. On the basis of these confident observations, the observer assigns a high value to $V(t)$

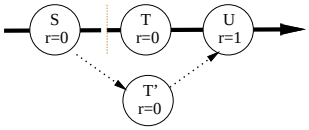

Figure 2: Prior Guidance

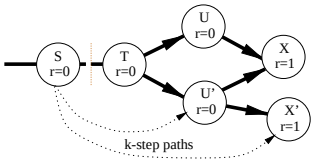

Figure 3: Reachability

and $V(s)$ and the observer is thereby encouraged to move toward these states during exploration. But suppose that after some time the mentor's action at state t is judged to be infeasible (e.g., there is an obstacle navigable by the mentor but not the learner). Unless the observer has embarked on sufficient exploration in the area to discover an alternate path from s to u (eg. through t') before the judgement, the value of state s will plunge immediately which will in turn eliminate the observer's future motivation to move towards state s and explore local alternatives from that point. If, however, the observer assumes by default that it has a roughly similar trajectory to that of the mentor, it may persist in backing up value from t to s in the belief that it will be able to discover a "local" path or *bridge* from s to u. Intuitively, a bridge is a "short" feasible path which bridges the gap in the value function due to an infeasible action. It starts on the mentor's trajectory in the state where the observer cannot duplicate the mentor's action and then navigates around the infeasible transition before ending on a state also on the mentor's trajectory but downstream of the infeasible transition. Such bridges can provide important guidance in cases where the value at a state (as defined by the augmented Bellman backup) is determined by the mentor's action rather than the learner's own actions. At such states, value estimates drop drastically as soon as the mentor's action is discovered to be infeasible unless a bridge has been discovered.

We note that bridges are often formed naturally in the imitation model as formulated thus far. Given a uniform prior over possible action effects,[4] each state is judged initially to be "reachable" with nonnegligible probability from states in its neighborhood. When a situation occurs (as described above) in which the mentor's action at s is deemed infeasible, the learner's value estimate $V(s)$ drops. However, this drop is often mitigated by the "flow" of value around the obstacle through neighboring states (e.g., t'). The use of uniform priors often seems to help this process along. This will encourage the learner to persist in exploring this neighborhood—thus, if a feasible bridge exists, it is likely to be found fairly early.

Prior guidance is not a reliable means of discovering bridges however. The combined effect of discounting and the small prior probability of state transitions cause the amount of value backed up to decrease very rapidly with the length of the trajectory along which is being backed up. Any negative rewards present can easily drown out small values. Thus at states at any significant distance from s, the value gradient is unlikely to point toward s in a significant way. We therefore consider a more explicit means of encouraging exploration in the area. Our *k-step repair strategy* initiates explicit searches for bridges, specifying explicit criteria for detecting their formation and caching the existence of a bridge in order to eliminate the need to check for it in the future.

k-step repair uses reachability analysis (based on the learner's current domain model) to test for the existence of a bridge. Consider the situation in Figure 3. When the learner first discovers that that the mentor's action at state s is infeasible, it undertakes a search for an existing bridge. Let a *bridge termination state* be any state on the mentor's trajectory within the k steps following state s. The observer now searches for a bridge, also k steps long which starts at state s, follows only feasible transitions and terminates in a bridge termination state. Because only feasible transitions are considered, misleading priors do not have undue influence. If a bridge is found, the mentor's influence is ignored at state s as value to should already be "flowing" back through the existing bridge. We flag the state as bridged so that we will not have to perform the bridge test again.

If a bridge is not found, however, we do not immediately suppress the mentor's influence at this state. Intuitively, we keep value flowing back to encourage the observer to come to the state with an infeasible action and explore the local neighbourhood before discounting the mentor's influence. If imitation is sensible in a given domain, we expect that it will be reasonable to assume that the path can be repaired by a short search of k-steps. The search is performed by a k^2-step random walk (in our 2-D grid worlds), which on average explores locations out to k-steps from the starting point (but not all locations up to k-steps away from s [18]). If during this walk the observer encounters a bridge termination state, we set the bridge-found flag for the originating state and suppress the value backup over the infeasible transition.[5] Attempts to discover bridges (as long

[4] We exploit local topology in our grid world experiments, so that a state is connected by any action *a priori* to its eight neighbours and to itself.

[5] Since actions are stochastic, there is no guarantee that executing the correct action to form a bridge at any given state will in fact perform the required transition to connect the bridge. Of course, even if a bridge is discovered, there is no guarantee that it is the optimal bridge. For the present, we will accept this possibility, with the understanding that any bridge will increase the attractiveness of state s.

as a bridge remains undiscovered) are performed n times (i.e., at n visits to state s). During this time, suppression of the mentor's influence is itself suppressed. After n random walks, no more attemtps are made, and the mentor's influence at state s is suppressed once and for all.[6]

4.1 Integrating Feasibility, k-step repair and Implicit Imitation

Feasibility and k-step repair can be easily integrated into the existing imitation framework. The complete decision procedure appears in Figure 4. As in the original model, we first check to see if the observer's experience-based calculation for the value of the state is more confident than the mentor-based calculation; if so, then the observer uses its own experience-based calculation. Otherwise, we check to see if the observer has a sufficient number of samples of its own behaviour to perform an action feasbility test. If not, we assume by default that the action taken by the mentor is feasible for the observer. This assumption will cause no permanent harm, as an error can only increase the value of the state which will in turn cause the observer to explore the state and increase the number of experience-based samples it has for this state. We currently use a threshold of 5 samples.

If there are a sufficient number of samples of the observer's actions, we perform the action feasibility test. If the mentor's action is feasible, then we accept the more confident value calculated using the mentor-observations based value function. If the action is infeasible we check to see if it is possible to do more bridging. The test checks two qualities of the state: If a bridge is already built then bridging is unnecessary. If we have exhausted our threshold for bridging attempts we say that it is impossible. In either case, no bridging actions are necesssary so we can dispense with mentor guidance and use the observer's own experience based calculations. If bridging is still possible then we delay suppression of mentor influence so that the augmented value function will guide the agent to the bridge building states and a repair can potentially be made.

5 Empirical Demonstrations

In this section, we empirically demonstrate the utility of feasibility testing and k-step repair and show how the techniques can be used to surmount both differences in actions between agents and small local differences in state-space topology.

In the first test, we show the necessity for feasibility testing in implicit imitation when agents have heterogeneous actions. In this scenario, all agents are confronted with the problem of navigating across an obstacle-free gridworld

[6] One can determine a "suitable" level of persistance (i.e., threshold n) using assumptions about the state space structure and noise level of actions. E.g., $n > 8k - 4$ seems suitable in an 8-connected grid world with low noise.

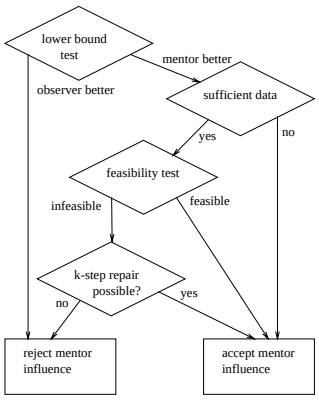

Figure 4: Implicit Imitation with Feasibility Tests

from the upper left corner to a goal location in the lower right where they are rewarded for their trouble. The agent is then reset to the top corner. We define a mentor which employs the "NEWS" set of actions which includes North, South, East and West. The mentor is trained and has an efficient and stationary policy for navigating from the start to the goal. A second agent employing implicit imitation with feasibility testing observes the mentor and uses these observations to backup an augmented value function. The observer, has a different set of actions called "Skew" which includes "North, South, North-East and South-West". This action set is significantly different from the "NEWS" set but doesn't alter the tendency of the agent to find the goal location in our test problems. In order to duplicate one of the mentor's "East" actions , the observer would have to perform a "North-East" action followed by a "South" action. This means that the observer's task is more difficult than the mentor's task. A third agent was introduced to act as a control for feasibility testing. It has the same "Skewed"actions as the first observer and also observes the same mentor but it does *not* use feasibility testing. A fourth agent was introduced to act as a control for all forms of imitation. It uses the "Skew" action set and it attempts to solve the same problem as the observers without the implicit imitation mechanism. All agents experience some stochastic effects in their actions which is modelled by perturbing the agent's action choice randomly 5% of the time. As in [14] all agents use model-based reinforcement learning with prioritized sweeping [12].

The results of the experiment are plotted as graphs. The horizontal axis represents time in simulation steps. The vertical axis represents the average number of goals achieved per 1000 time steps. Examining the graph in Figure 5 we see that the imitation agent with the benefit of implicit imitation and action feasibility testing is the first to find the goal and starts converging towards optimal goal rate well before the other agents. The agent that attempts to apply implicit imitation without feasibility testing achieves sporadic success early on, but frequently "locks up" due to repeated attempts

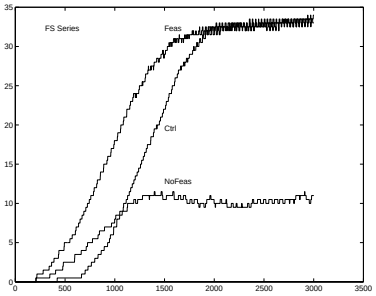

Figure 5: Utility of Feasibility Testing

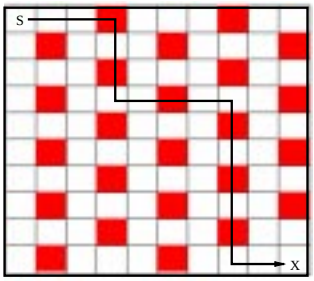

Figure 6: Obstacle Map and Mentor Path

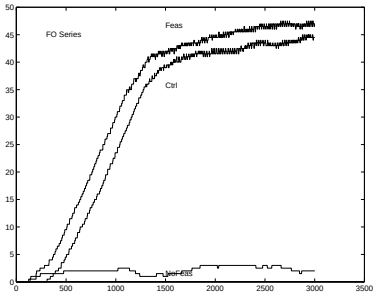

Figure 7: Interpolating Around Obstacles

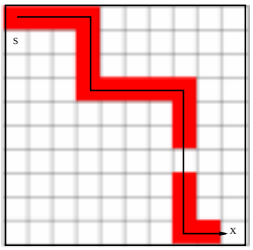

Figure 8: Parallel Generalization

to duplicate infeasible mentor actions. The agent still manages to reach the goal since the stochastic actions do not permit the agent to become permanently stuck in the obstacle-free scenario but its convergence stalls well before reaching the optimal goal rate. The control agent without any form of imitation demonstrates a significant delay in convergence relative to the imitation agents due to the lack of any form of guidance, but easily surpasses the agent without feasibility testing in the long run. We can therefore see that feasibility testing is necessary in situations with heterogeneous actions. We do not have sufficient space to present the results here, but we note that the gains due to imitation with feasibility testing increase with problem size and difficulty.

We developed feasibility testing and bridging primarily to deal with the problem of adapting to agents with heterogeneous actions. The same techniques, however, can be applied to agents with differences in their state spaces. Again, we assume that the imitation agents are given the mapping from the mentor's space to their own. In this case, however, we give the imitation agent a different environment than that of the mentor. In Figure 6 we see the obstacles in the observer's environment and the path of the mentor is marked by the arrow drawn over top.

The heterogeneous action model is sufficient to deal with the problem however, as these differences in state space "look like" actions that have different effects. When the imitator attempts to move into an obstacle, his action does something different than the mentor's. In Figure 7 we see that the results are similar to the previous case with different actions. Here, however, the top goal rate achieved by the observer with feasibility testing and the control agent is much higher because there is a path the same length as the mentor's optimal path, but it is a different path. The observer without feasibility has a more difficult time with this maze as the physical obstacles make it more difficult for the agent to achieve the goal purely by advancing due to the stochastic randomness of its actions. It has almost no success.

Next we demonstrate how feasibility testing can completely generalize the mentor's trajectory. Here the mentor follows a path which is completely infeasible for the imitating agent. To demonstrate this, we fix the mentor's path for all runs and then we give the imitating agent a maze shown in Figure 8 in which all but two of the states the mentor visits are blocked by an obstacle. The imitating agent is able to use the mentor's trajectory for guidance and builds its own parallel trajectory which is completely disjoint from the mentor's.

The results in Figure 9 require some explanation. In this experiment the environment has two large weakly connected regions. This type of feature in a scenario induces high variance in the results. When the control agent happens to find the doorway between the two regions early on, it will often find the goal state shortly after. The exact time of this discovery varies significantly from run to run. When the runs are averaged, it causes the steps in the graph. On average, the control agent's performance is worse than the imitation agent with feasibility. The agent without feasibility testing does very poorly. This is because it gets stuck

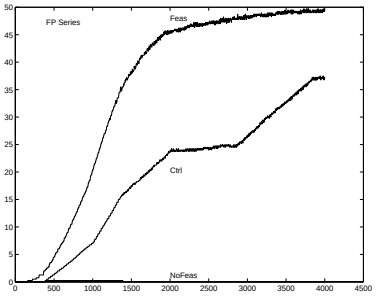

Figure 9: Parallel Generalization Results

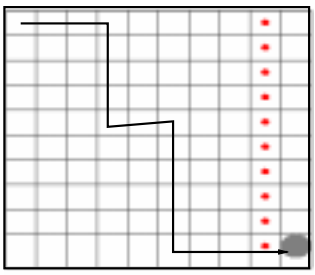

Figure 10: River Scenario

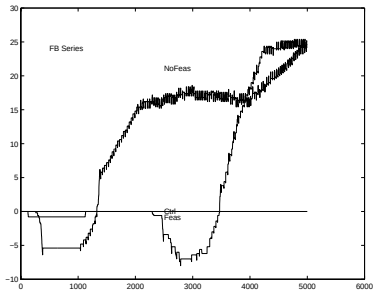

Figure 11: Utility of Bridging

around the doorway. The high value gradient backed up along the mentor's path becomes accessible to the agents at the doorway. The imitation agent with feasibility will eventually conclude that it cannot proceed south from the doorway to follow the mentor's gradient and try a different strategy. The imitator without feasibility testing never explores far enough away from the doorway to setup an independent value gradient that will guide it to the goal. With a slower decay schedule for exploration, the imitator without feasibility testing would find the goal, but this would still reduce its performance below that of the imitator with feasibility testing. The imitator with feasibility testing makes use of its prior beliefs that it can follow the mentor to backup value perpendicular to the mentor's path. An aura of value thus clings to the mentor's path and the imitator can rapidly follow this aura to the doorway, make the necessary feasibility test at the doorway and the proceed to the goal.

5.1 Utility of Bridging

As explained earlier, in simple problems there is a good chance that the informal effects of prior value leakage and stochastic exploration may form bridges before feasibility testing cuts off the value propagation that guides exploration. In more difficult problems where the agent spends a lot more time exploring, before finding the goal, it will have sufficient samples to conclude that the mentor's actions are infeasible long before the agent has constructed its own bridge. While losing the mentor's guidance after several trials of an action causes no real penality to the agent over a control agent trying to solve the problem, we would still like to see an imitation agent make use of the mentor's trajectory in order to do better.

Our test environment, pictured in Figure 10 will be the scenario we call the river maze. Agents must navigate from the top-left corner to the bottom-right by passing through a river running vertically through the maze. The goal state is worth +1.0, but each river state crossed exacts a penalty of -0.2. Without a long exploration phase, agents generally can't reach the goal, because they find the negative penalties of the river and develop a policy to avoid the river without exploring beyond it. The negative penalties also tend to discourage forward exploration limiting the potential for forward bridges to form spontaneously.

While we had hoped that this scenario would reveal a definite advantage for agents with k-step repair, the results have not been conclusive. In Figure 11, we see that both imitation agents perform better than the control, but k-step repair doesn't offer a big advantage to one imitation agent over the other. In this graph the "NoFeas" curve represents the agent without k-step repair. We are just analyzing our data and checking for bugs now and hope to have this ready for the final paper.

6 Conclusion

We have seen that feasibility testing extends implicit imitation in a principled manner to deal with the situations where the homogeneous actions assumption proves to be invalid. Adding bridging capabilities preserves and extends the guidance mentor's can give observers in the presence of infeasible actions whether due to differences in action capabilities or local differences in state spaces. Our new approach makes use of a model to compute the actions an imitator should take without requiring that the observer duplicate the mentor's actions exactly. Our approach also relates to the idea of "following" in the sense that the imitator uses local search in its model to repair discontinuities in its augmented value function before acting in the world.

We see two major directions for future development on this line of research. The first would be apply the implicit imitation with feasibility testing and k-step repair to some interesting applications. We expect that combining our en-

hanced algorithm with more advanced exploration techniques and some generalization capabilities will open up a broad range of applications such as mobile robot navigation, plant control, language learning and others. We would choose for our second direction, to look at augmenting our model with capabilities to reason in partially observable environments and perhaps make explicit use of abstraction techniques.

References

[1] C. G. Atkeson and S. Schaal. Robot learning from demonstration. *ICML-97*, 1997.

[2] Paul Bakker and Yasuo Kuniyoshi. Robot see, robot do : An overview of robot imitation. *AISB96 Workshop on Learning in Robots and Animals*, pages 3–11, 1996.

[3] Aude Billard and Gillian Hayes. Learning to communicate through imitation in autonomous robots. *ICANN 97*, pages 763–68, 1997.

[4] Kerstin Dautenhahn. Getting to know each other – artificial social intelligence for autonomous robots. *Robotics and Autonomous Systems*, 16:333–356, 1995.

[5] Leo A. Goodman. On simultaneous confidence intervals for multinomial proportions. *Technometrics*, 7(2):247–54, 1965.

[6] J Hayes, GM; Demiris. Robotic learning by imitation. In *The 3rd International Conference on Simulation of Adaptive Behavior, From Animals to Animats,*, UK, 1994.

[7] Leslie Pack Kaelbling. *Learning in Embedded Systems*. MIT Press, Cambridge, 1993.

[8] Yasuo Kuniyoshi, Masayuki Inaba, and Hirochika Inoue. Learning by watching: Extracting reusable task knowledge from visual observation of human performance. *IEEE Transactions on Robotics and Automation*, 10(6):799–822, 1994.

[9] Maja J. Mataric. Using communication to reduce locality in distributed multi-agent learning. *Journal of Experimental and Theoretical Artificial Intel.*, 10(3):357–369, 1998.

[10] J. Mi and Allan R. Sampson. A comparison of the bonferroni and scheffé bounds. *Journal of Statistical Planning and Inference*, 36:101–105, 1993.

[11] T. M. Mitchell, S. Mahadevan, and L. Steinberg. LEAP: A learning apprentice for VLSI design. *IJCAI-85*, pages 573–580, 1985.

[12] Andrew W. Moore and Christopher G. Atkeson. Prioritized sweeping: Reinforcement learning with less data and less real time. *Machine Learning*, 13, 1993.

[13] Chrystopher Nehaniv and Kerstin Dautenhahn. Mapping between dissimilar bodies: Affordances and the algebraic foundations of imitation. In *Proc. European Workshop on Learning Robots (EWRL-7)*, pages 64–72, Edinburgh, 1998.

[14] Bob Price and Craig Boutilier. Implicit imitation in multiagent reinforcement learning. In Ivan Bratko and Saso Dzeroski, editors, *Machine Learning: Proeedings of the Sixteenth Internation Conference (ICML '99)*, pages 325–34. Morgan Kaufmann Publishers, Inc., 1999.

[15] Henry Scheffe. *Analysis of Variance*. Wiley, New York, 1959.

[16] George A. F. Seber. *Multivariate Observations*. Wiley, New York, 1984.

[17] Paul E. Utgoff and Jeffrey A. Clouse. Two kinds of training information for evaluation function learning. *Proceedings of Ninth National Conference on Artificial Intelligence*, pages 596–600, 1991.

[18] Eric W. Weisstein. *CRC Concise Encyclopedia of Mathematics*. CRC Press, 1996. http://mathworld.wolfram.com/.