Imitation and Reinforcement Learning in Agents with Heterogeneous Actions

Bob Price[1] and Craig Boutilier[2]

[1] Department of Computer Science, University of British Columbia, Vancouver, B.C., Canada V6T 1Z4 `price@cs.ubc.ca`
[2] Department of Computer Science, University of Toronto, Toronto, ON, Canada M5S 3H5 `cebly@cs.toronto.edu`

Abstract. Reinforcement learning techniques are increasingly being used to solve difficult problems in control and combinatorial optimization with promising results. *Implicit imitation* can accelerate reinforcement learning (RL) by augmenting the Bellman equations with information from the observation of expert agents (mentors). We propose two extensions that permit imitation of agents with heterogeneous actions: feasibility testing, which detects infeasible mentor actions, and k-step repair, which searches for plans that approximate infeasible actions. We demonstrate empirically that both of these extensions allow imitation agents to converge more quickly in the presence of heterogeneous actions.

1 Introduction

Traditional methods for solving difficult control and combinatorial optimization problems have made frequent recourse to heuristics to improve performance. Increasingly, adaptive methods such as reinforcement learning have been used to allow programs to learn their own heuristic or "value" functions to guide search. The results in such diverse areas as job-shop scheduling [1] and global optimization problems [2] have been quite promising. Typically, however, the types of problems we would like to solve are similar to problems already solved or to problems being pursued by others. We have therefore argued [3], as have others, for a broader, sociologically inspired model of reinforcement learning which can incorporate the knowledge of multiple agents solving multiple related problems in a loosely coupled way.

Coupling between agents is typically achieved through communication, however, the lack of a common communication protocol or the presence of a competitive situation can often make explicit communication infeasible. We have demonstrated, using simple domains, that it is possible to overcome communication barriers by equipping agents with imitation-like behaviors [3]. Using imitation, agents can learn from others without communicating an explicit context for the applicability of a behavior [4]; without the need for an existing communication protocol; in competitive situations where agents are unwilling to share information; and even when other agents are unwilling to fulfill a teacher role. The ability of imitation to effect skill transfer between agents has also been demonstrated in a range of domains [5–10]. These domains, however, have dealt with agents imitating other agents with similar actions. Our goal is to extend imitation to allow agents to learn from expert agents (mentors) with different action capabilities or inhabiting different environments. For example, an agent learning to control a newly upgraded elevator group in a large building could benefit from the adaptive learning of a prior controller on the previous elevator system of that building.

Previously, we have showed that *implicit imitation* can accelerate reinforcement learning (RL) by allowing agents to take advantage of the knowledge implicit in observations of more skilled agents [3]. Though we did not assume that the learner shared the same objectives as the mentors, we did rely on the fact that actions were *homogeneous*: every action taken by a mentor corresponded to some action of the learner. In this work, we relax this assumption and introduce two mechanisms that allow acceleration of RL in presence of *heterogeneous actions*: *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.

Our work can be viewed loosely as falling within the framework proposed by Nehaniv and Dautenhahn [11], who view imitation as the process of constructing mappings between states, actions, and goals of different agents (see also the abstraction model of Kuniyoshi at al. [8]). Unlike their model, we assume that state-space mappings are given, the mentor's actions are not directly observable, the goals of the mentor and learner may differ, and that environments are stochastic. Furthermore, we do not require that the learner explicitly duplicate the behavior of the mentor. Our model is also related to behavioral cloning, but again we do not share the goal of behavioral cloning which aims to reproduce an observed behavior by inducing an objective function from observed behavior [12]. As in [5], our model incorporates an independent learning and optimization component that differs from "following" and "demonstration" models often used in robotics [7, 13], though the repair strategies we invoke do bear some relation to "following" models.

2 Imitation with Homogeneous Actions

In this section we summarize the implicit imitation model developed in [3]. Further details and motivation can be found in this paper. In *implicit imitation* [3], we assume two agents, a *mentor* m and an *observer* o, acting in a fixed environment.[1] We assume the observer (or learner) is learning to control a Markov decision process (MDP) with states S, actions 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).

[1] The extension to multiple mentors with varying expertise is straightforward [3].

We make two assumptions: the mentor implements a deterministic stationary policy π_m, which induces a Markov chain $\Pr_m(t|s) = \Pr_m(t|s, \pi_m(s))$ over S;[2] and for each action $\pi_m(s)$ taken by the mentor, there exists an action $a \in A_o$ such that the distributions $\Pr_o(\cdot|s,a)$ and $\Pr_m(\cdot|s)$ 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 the mentor's action $\pi_m(s)$ (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, namely, 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.[3] 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 [3], 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.

At states the mentor visits infrequently (because they are rarely traversed by its optimal policy), the learner's estimates of the mentor's Markov chain may be poor 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 [15] interval estimation method, we use the variance in our estimated Dirichlet distributions for the model parameters to construct crude lower bounds on both the augmented value function incorporating the mentor model and an unaugmented value function based strictly on the observer's own experience. If the lower bound on the augmented value function is less than

[2] Generalization to stochastic policies can easily be handled.

[3] We assume a model-based RL algorithm (e.g., prioritized sweeping [14] and an exploration model which is influenced by state values (e.g. ϵ greedy).

Table 1. Augmented Backup

```
FUNCTION augmentedBackup(V+, Pr_o, σ_o^2, Pr_m, σ_m^2, s)
```

$a^* = \mathrm{argmax}_{a \in A_o} \sum_{t \in S} Pr(s,a,t)V^+(t)$

$V_o(s) = R_o(s) + \gamma \sum_{t \in S} Pr(s,a^*,t)V(t); \quad V_m(s) = R_o(s) + \gamma \sum_{t \in S} Pr_m(s,t)V(t)$

$\sigma_o(s) = \gamma^2 \sum_{t \in S} \sigma(s,a^*,t)V^+(t)^2; \quad \sigma_m(s) = \gamma^2 \sum_{t \in S} \sigma_m(s,t)V^+(t)^2$

$V_o^-(s) = V_o(s) - \sigma_o(s); \quad V_m^-(s) = V_m(s) - \sigma_m(s)$

IF $V_o > V_m$ THEN $V^+(s) = V_o(s)$
ELSE $V^+(s) = V_m(s)$
RETURN $V^+(s)$

the lower bound on the unaugmented value function, then either the augmented value is in fact lower, or it is highly variable. Using lower bounds ensures that uncertainty about an action model makes it look worse. In either circumstance, suppression of the mentor's influence is appropriate and we use an unaugmented Bellman backup.

In the algorithm shown in Table 1, the inputs are the observer's augmented value function V^+, its action model and variance PR_o, σ_o^2, the action model and variance for mentor observations Pr_m, σ_m^2 and the current state s. The output is a new augmented value for state s. The program variable $V_o(s)$ represents the best value the observer can obtain in state s using its own experience-based action models and $V_m(s)$ represents the value the agent could obtain if it employed the same action as the mentor. The term $\sigma_o^2(s)$ represents a conservative overestimate of the variance in the estimate of the value of state s, $V(s)$, due to the *local* model uncertainty in the observer's own action models and $\sigma_m^2(s)$ represents a similar uncertainty in the estimate derived from the mentor action model. The uncertainty is used to construct loose lower bounds on the value estimates denoted V_o^- and V_m^-. These bounds are crude but sufficient to suppress mentor influence at appropriate states.

3 Imitation with Heterogeneous Actions

When the homogeneity 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 (with the same probability) as the mentor at a state s, it may drastically overestimate the value of s.[4] The inflated value estimate may cause the learner to return repeatedly to this (potentially undesirable) state with a potentially drastic impact on convergence time

[4] Augmented backups cannot cause underestimation of the value function.

(see Section 4). Implicit imitation has no mechanism to remove the unwanted influence of the mentor's model (confidence estimates play no role here). What is needed is the ability to identify when the key assumption justifying augmented backups—that the observer can duplicate *every* mentor action—is violated.

In such heterogeneous settings, this issue can be resolved by the use of an explicit *action feasibility test*: before an augmented backup is performed at s, the observer tests whether the mentor's action a_m "differs" from each of its actions at s, given its current estimated models. If so, the augmented backup is suppressed and a standard Bellman backup is used to update the value function. By default, mentor actions are assumed to be feasible for the observer; however, once the observer is reasonably confident that a_m is infeasible at state s, augmented backups are suppressed at s.

Recall that action models are estimated from data with the learner's uncertainty about the true transition probabilities reflected in a Dirichlet distribution. Comparing a_m with a_o is effected by a difference of means test w.r.t. the corresponding Dirichlets. This is complicated by the fact that Dirichlets are highly non-normal for small sample counts. We deal with the non-normality by requiring a minimum number of samples and using robust Chebyshev bounds on the pooled variance of the distributions to be compared. When we have few samples, we persist with augmented backups (embodying our default assumption of homogeneity). If the value estimate is inflated by these backups, the agent will be biased to obtain additional samples which will then allow the agent to perform the required feasibility test. We deal with the multivariate complications by performing the *Bonferroni test* [16] which has been shown to give good results in practice [17], is efficient to compute, and is known to be robust to dependence between variables. A Bonferroni hypothesis test is obtained by conjoining several single variable tests. Suppose the actions a_o and a_m result in r possible outcomes, $s_1, \cdots, s_r$, at s (i.e., r transition probabilities to compare). For each s_i, hypothesis E_i denotes that a_o and a_m have the same transition probability $\Pr(s_i|s)$, and $\bar{E}_i$ the complementary hypothesis. The Bonferroni inequality states:

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

Thus we can test the joint hypothesis $\bigcap_{i=1}^{r} E_i$—the two action models are the same—by testing each of the r complementary hypotheses $\bar{E}_i$—transition probability for outcome i is the same— at confidence level α/r. If we reject any of the complementary hypotheses we reject the notion that the two actions are equal with confidence α. The mentor action a_m is deemed infeasible if for every observer action a_o, the multivariate Bonferroni test, just described, rejects the hypothesis that the action is the same as the mentor's.

The feasibility test is summarized in Table 2. The feasibility test tests whether the action demonstrated by mentor m in state s, is likely to be feasible for the observing agent in state s. The parameters of the observer's own Dirichlet distributions are denoted $n_o(s, a, t)$ which denotes the number of times the observer observes itself making the transition from state s to state t when it executes action a in state s. The parameters for the mentor action model are denoted $n_m(s,t)$ which gives the number of times the observer observes the mentor making the transition from state s to state t. The difference of means is denoted μ_Δ and the test statistic z_Δ.

Table 2. Action Feasibility Testing

```
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(s,a,t) − Pr_m(s,t)
         z_Δ = μ_Δ sqrt( (n_o(s,a,t)*var_o(s,a,t)+n_m(s,t)var_m(s,t)) / (n_o(s,a,t)+n_m(s,t)) )
         IF z_Δ > z_α/r
            allSuccessorProbsSimilar = false
      END FOR
      IF allSuccessorProbsSimilar THEN return true
   END FOR
   RETURN false
```

Action feasibility testing has some unintended effects. Suppose an observer has previously constructed an estimated value function using augmented backups. Subsequently, the mentor's action a_m is judged to be infeasible at s. If the augmented backup is suppressed, the value of $V(s)$ and all of its preceding states will drop as value backups propagate the change through the state space. As a result, the bias of the observer toward s will be eliminated. However, imitation is motivated by the fact that the observer and mentor are similar is some respects. We might hope, therefore, that there exists a short path or a *repair* around the infeasible transition. The observer's ability to "duplicate" a_m might take the form of local policy rather than a single action.

To encourage the learner to explore the vicinity of an infeasible action, we will sometimes consider retaining the mentor's influence through augmented backups and then use the notion of *k-step repair* to search for a local policy. Specifically, when a_m is discovered to be infeasible at state s, the learner undertakes a k-step reachability analysis (w.r.t. its current model Pr_o) to determine if it can "workaround" the infeasible action (i.e., find a k-step path from s to a point on the mentor's nominal trajectory). If so, the learner knows that value will "flow" around the infeasible transition and thereby maintain the existing exploration bias. In this case, the learner concludes that the state is already "repaired" and augmented backups are suppressed. Otherwise, a random walk with expected radius of k-steps is undertaken to explore the area. This allows the learner to improve its model and discover potential repair paths. This walk is repeated at the next n visits of s or until a repair path is found. If no repair is found after

n attempts, the agent concludes that the infeasible transition is irreparable and augmented backups are suppressed permanently. Thus the mentor's influence persists in guiding the learner toward s until it is deemed to be unnecessary or misleading. The parameters k, and n must be tuned empirically, but can be estimated given knowledge of the connectivity of the domain and prior beliefs about how similar (in terms of length of average repair) the trajectories of the mentor and observer will be.

4 Empirical Demonstrations

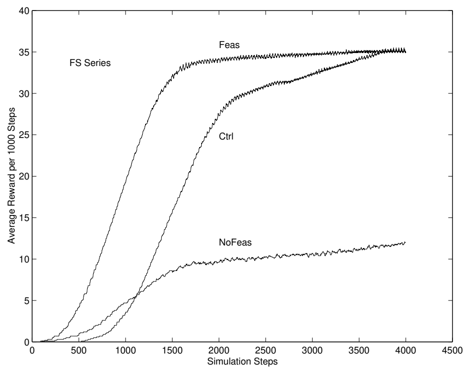

Fig. 1. Utility of Feasibility Testing

Experimental evaluation of the original implicit imitation mechanism can be found in [3]. Our first experiment in this paper illustrates the necessity of feasibility testing. Agents must navigate an obstacle-free, 10-by-10 grid-world from upper-left corner to lower-right. We give a mentor with the "NEWS" action set (North, South, East and West movement actions) an optimal stationary policy. We study three learners, with the "Skew" action set (N, S, NE, SW) which are unable to duplicate the mentor exactly. The first learner imitates *with* feasibility testing, the second *without* feasibility testing, and the third control agent uses no imitation (i.e., is a standard RL-agent). Actions are perturbed 5% of the time. As in [3] the agents use model-based reinforcement learning with prioritized sweeping [14]. We used $k = 3$ and $n = 20$.

In Figure 1 the horizontal axis represents time and the vertical axis represents the average reward per 1000 time steps (averaged over 10 runs). The imitation agent with feasibility testing converges quickly to the optimal rate. The agent without feasibility testing achieves sporadic success early on, but due to frequent attempts to duplicate infeasible actions it never converges to the optimal rate (stochastic actions permit it to achieve goals eventually). The control agent without guidance due to imitation demonstrates a delay in convergence relative to the imitation agents, but converges to optimal rate in the long run. The gradual slope of the control agent is due to the higher variance in the control agent's discovery time for the optimal path. Thus, we see that imitation improves convergence, but feasibility testing is necessary when heterogeneous actions are present.

We developed feasibility testing and k-step repair to deal with heterogeneous actions, but the same techniques can be applied to agents operating in state space with different connectivity (these are equivalent notions ultimately). We constructed a domain where all agents have the *same* NEWS action set; but we introduce obstacles as shown in Figure 2, into the environment of the learners. The obstacles cause the imitator's actions to have different effects than the mentor's.

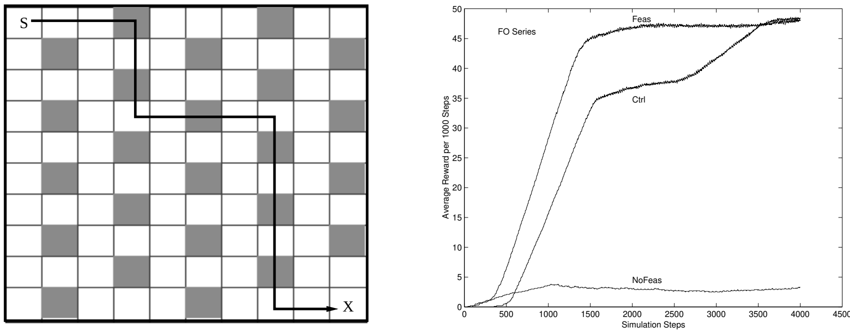

Fig. 2. Obstacle Map, Mentor's Path and Experiment Results

In Figure 2 we see that the imitator with feasibility testing performs best, the control agent converges eventually, and the agent without feasibility testing stalls. The optimal goal rate is higher in this scenario because the agents use the same "efficient" NEWS actions. We see local differences in connectivity are well handled by feasibility testing.

In simple problems it is likely that a learner's exploration may form possible repair paths before feasibility testing cuts off the guidance obtained from mentor observations. In more difficult problems (e.g., where the learner spends a lot of time exploring), it may conclude that a mentor's action is infeasible long before it has constructed its own repair path. The imitator's performance would then drop down to that of an unaugmented reinforcement learner.

To illustrate the effectiveness of k-step repair, we devised a domain where agents must cross a three-step wide "river" which runs vertically and exacts a penalty of -0.2 per step (see Figure 3). The goal state is worth +1.0. Without a long exploration phase, agents generally discover the negative states of the river and curtail exploration in this direction before actually making it across. If we examine the value function estimate (after 1000 steps) of an imitator with feasibility testing but no repair capabilities, we see that, due to suppression

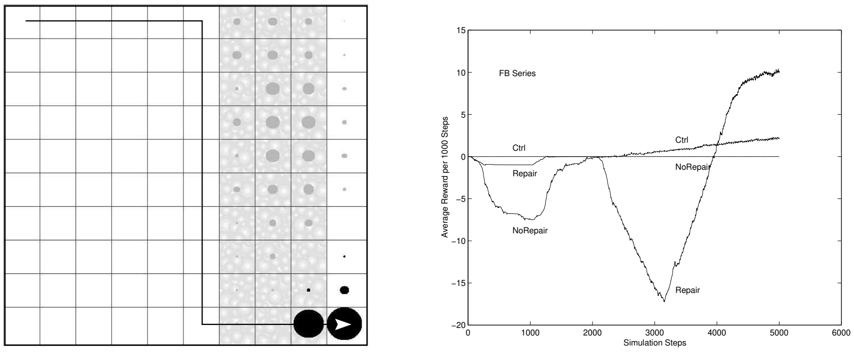

Fig. 3. River Scenario, Mentor's Path and Experiment Results

by feasibility testing, the high-value states (represented by large dark circles in Figure 3), backed up from the goal terminate abruptly at an infeasible transition before making it across the river. In fact, they are dominated by the lighter grey circles showing negative values. Once this barrier forms, only an agent with a very optimistic exploration policy will get to the goal, and then only after considerable exploration. In this experiment, we apply a k-step repair agent to the problem with $k = 3$.

Examining the graph in Figure 3, we see that both imitation agents experience an early negative dip as they are guided deep into the river by the mentor's influence. The agent without repair eventually decides the mentor's action is infeasible, and thereafter avoids the river (and the possibility of finding the goal). The imitator with repair also discovers the mentor's action to be infeasible, but does not immediately dispense with the mentor's guidance. It keeps exploring in the area of the mentor's trajectory using random walk, all the while accumulating a negative reward until it suddenly finds a repair path and rapidly converges on the optimal solution.[5] The control agent discovers the goal only once in the ten runs.

5 Conclusion

Implicit imitation makes use of the observer's own reward function and a model augmented by observations of a mentor to compute the actions an imitator should take without requiring that the observer duplicate the mentor's actions exactly. We have seen that feasibility testing extends implicit imitation in a principled manner to deal with the situations where the homogeneous actions assumption is invalid. Adding k-step repair preserves and extends the mentor's guidance in the presence of infeasible actions, whether due to differences in action capabilities or local differences in state spaces. Our approach also relates to the

[5] While repair steps take place in an area of negative reward in this scenario, this need not be the case. Repair doesn't *imply* short-term negative return.

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 are currently extending our model to deal with partially-observable environments and to make explicit use of abstraction and generalization techniques in order to tackle a wider range of problems.

References

1. Wei Zhang and Thomas G. Dietterich. A reinforcement learning approach to job-shop scheduling. In *IJCAI-95*, pages 1114–1120, Montreal, 1995.
2. Justin A. Boyan and Andrew W. Moore. Learning evaluation functions for global optimization and boolean satisfiability. In *AAAI-98*, pages 3–10, July 26-30, 1998, Madison, Wisconsin, 1998.
3. Bob Price and Craig Boutilier. Implicit imitation in multiagent reinforcement learning. In *ICML-99*, pages 325–334, Bled, SI, 1999.
4. Paul Bakker and Yasuo Kuniyoshi. Robot see, robot do : An overview of robot imitation. In *AISB96 Workshop on Learning in Robots and Animals*, pages 3–11, Brighton,UK, 1996.
5. C. G. Atkeson and S. Schaal. Robot learning from demonstration. In *ICML-97*, pages 12–20, Nashville, TN, 1997.
6. Aude Billard and Gillian Hayes. Learning to communicate through imitation in autonomous robots. In *ICANN-97*, pages 763–68, Lausanne, Switzerland, 1997.
7. G. M. Hayes and J. Demiris. A robot controller using learning by imitation. Technical Report DAI No. 676, University of Edinburgh. Dept. of Artificial Intelligence, 1994.
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. T. M. Mitchell, S. Mahadevan, and L. Steinberg. LEAP: A learning apprentice for VLSI design. In *IJCAI-85*, pages 573–580, Los Altos, California, 1985. Morgan Kaufmann Publishers, Inc.
10. Paul E. Utgoff and Jeffrey A. Clouse. Two kinds of training information for evaluation function learning. In *AAAI-91*, pages 596–600, Anaheim, CA, 1991. AAAI Press.
11. Chrystopher Nehaniv and Kerstin Dautenhahn. Mapping between dissimilar bodies: Affordances and the algebraic foundations of imitation. In *EWLR-98*, pages 64–72, Edinburgh, 1998.
12. Dorian Šuc and Ivan Bratko. Skill reconstruction as induction of LQ controllers with subgoals. In *IJCAI-97*, pages 914–919, Nagoya, 1997.
13. Maja J. Mataric, Matthew Williamson, John Demiris, and Aswath Mohan. Behaviour-based primitives for articulated control. In *SAB-98*, pages 165–170, Zurich, 1998.
14. Andrew W. Moore and Christopher G. Atkeson. Prioritized sweeping: Reinforcement learning with less data and less real time. *Machine Learning*, 13(1):103–30, 1993.
15. Leslie Pack Kaelbling. *Learning in Embedded Systems*. MIT Press, Cambridge,MA, 1993.
16. George A. F. Seber. *Multivariate Observations*. Wiley, New York, 1984.
17. J. Mi and Allan R. Sampson. A comparison of the Bonferroni and Scheffé bounds. *Journal of Statistical Planning and Inference*, 36:101–105, 1993.