NOTE TO USERS

Risk-Directed Exploration in Reinforcement Learning

Edith L. M. Law

Masters of Science

Department of Computer Science

McGill University

Montreal,Quebec

2005-02-21

A Thesis submitted to McGill University
in partial fulfillment of the requirements of the degree of
Master of Science in Computer Science

Your file Votre référence
ISBN: 0-494-12483-0
Our file Notre référence
ISBN: 0-494-12483-0

ACKNOWLEDGEMENTS

Over the course of two years, I was fortunate to be under the guidance of many people who shaped my thinking on this subject. First and Foremost, I thank my supervisor Dr. Doina Precup, for her insights, encouragement, and financial support throughout this time. The feedback she provided on this thesis and the many iterations of detailed editing allowed many concepts to be clarified and elaborated. The idea for this thesis came from my initial interests in computational models of emotion and how the associated behavior is related to survival. I benefited enormously from conversations with Dr. Jorge Armony, whose enthusiasm has inspired the continuance of my interests in the topic. I thank people in the Reasoning and Learning Lab for the thought-provoking discussions on the topic. Lastly, I want to thank Dr. Maria Klawe for believing in me and for having urged me to return to graduate school.

I also thank Juaned Sattar, Rafa Absar, Sumedha Ahuja, Seung-Hyun Park for their friendship, and especially Eric Blais, for juggling multiple roles in order for me to focus on my work, and for translating the abstract into French.

Lastly, I would like to thank my parents. Without their love and unrelenting support throughout the years, many of my endeavours in life would not be possible.

ABSTRACT

Reinforcement Learning is a class of methods for solving sequantial decision problems when the model of the environment is not known. In this framework, the agent must explore the environment to gather more information about the model and the utility of each of its actions, while striving to act as well as possible using limited knowledge. One of the major obstacles that prevent reinforcement learning from being extended to real-life settings is the fact that the agent is blind to the risk of actions during learning, potentially ending up in catastrophic states. This thesis presents a model-based directed exploration method for selecting actions based on a measure of risk, characterized by entropy and expected immediate reward. The weighted combination of this risk measure and the long term utility of the action, or *risk-adjusted utility*, is used to determine the probability of different actions. Using this approach, agents can manifest risk-averse or risk-seeking behavior. Experimental results show that risk-directed exploration can result in better performance during learning than the standard Boltzmann action selection method, or other directed exploration methods such as counter-based and recency-based methods.

ABRÉGÉ

L'apprentissage par renforcement est une classe de méthodes utilisée pour résoudre des problèmes de décisions séquentielles dans un environnement dont le modèle est inconnu. Dans ce cadre, l'agent doit explorer l'environnement pour accumuler plus d'information sur le modèle et la valeur de chaque action, tout en essayant d'agir de façon optimale à l'aide de ses connaissances acquises. Un des obstacles principaux qui nous empêche d'utiliser l'apprentissage par renforcement dans des situations réelles est le fait que l'agent n'est pas conscient des risques associés à ses actions pendant l'apprentissage. L'agent se retrouve donc régulièrement dans des états catastrophiques. Cette mémoire présente une méthode de sélection d'actions pour poursuivre une exploration dirigée de modèles. Cette méthode est basée sur une mesure de risque caractérisée par l'entropie et l'espérance de la valeur immédiate des actions. La combinaison pesée de cette mesure de risque et de la valeur á long terme de chaque action, ou *la valeur ajustée au risque*, est utilisée pour déterminer la probabilité de choisir les différentes actions. En utilisant cette approche, l'agent peut démontrer un comportement averse envers le risque ou de recherche de risque. Des résultats expérimentaux montrent que l'exploration dirigée par le risque peut donner de meilleurs résultats lors de l'apprentissage que la méthode de sélection d'action Boltzmann standard, ou que d'autres méthodes d'exploration dirigée telles que les méthodes basées sur des compteurs ou les méthodes basées sur la récence.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1
Introduction

Life is an error-making and an error-correcting process, and nature in marking man's papers will grade him for wisdom as measured both by survival and by the quality of life of those who survive.
- Jonas Salk

The creation of an intelligent agent that is able to carry out tasks autonomously requires major paradigm shifts in machine learning. First, instead of static environments, the agent typically operates in environments that are partially or completely unknown, uncertain, or changing. Second, the lack of prior knowledge about the environment means that having a provision of examples, as assumed in supervised learning, is not feasible. With limited knowledge and no teacher, the agent must continuously learn about the environment by selecting actions that are potentially suboptimal but informative (exploration), while at the same time, strive to behave optimally (exploitation).

This form of online learning, particularly in hazardous environments, poses a unique challenge. With incomplete knowledge of the environment, how can the agent survive by avoiding actions that lead to fatal consequences? The importance of this question is revealed when we consider the nature of some current applications of artificial intelligence. Intelligent agents are increasingly used for tasks that humans would not or could not perform. For example, robots may be deployed to collect data on an unexplored planet, clean up toxic waste at a disaster zone, and recover sunken objects from the deep sea. Artificial intelligence is used in medicine to make diagnosis, recommend short and long term treatment strategies,

or dynamically control biomedical devices. Evidently, in these examples of safety-critical systems, it is crucial for the system to be as conservative as possible, in order to minimize the risk of damaging expensive robotics machinery in the former case, and of undermining patient well-being in the latter. Bounding the overall risk of the system does not suffice, if the system is to be used in real time. The short term consequence of an action has to be weighted against the long term utilities of that action.

1.1 Reinforcement Learning in Real-life Settings

This thesis addresses the fundamental issue of self-preservation during online learning in uncertain environments within the context of reinforcement learning. Reinforcement learning is a class of computational methods which allow agents to learn how to behave optimally in a given environment. The goal is to learn a way of selecting actions that maximizes the long term expected return. If the model of the environment is unknown, the agent learns by taking actions in the environment, observing the reward signals it receives, and updating the utilities of actions in different states based on this experience. Learning takes place over a pre-specified number of episodes. In a hazardous environment, each episode can end either when the agent reaches the goal, or when it encounters a fatal state.

Several characteristics of reinforcement learning make it promising for autonomous agents. Few assumptions are made about the nature of the environment. This allows learning to take place in partially or completely unknown environments. Likewise, by allowing learning and execution to happen concurrently, the need for prior knowledge about the environment is eliminated.

Recently, there has been a lot of interest in the research community to extend reinforcement learning to real-life settings. This attempt, however,

is faced with great challenges (Bulitko, 2004). One of the most problematic issues is that learning takes place over multiple reincarnations of the agent. In real life, reincarnation is rarely an option. A less extreme version of this issue was raised also in the 2002 AAAI Spring Symposium on *safe learning agents* and the 2004 AAAI fall workshop on *real-life reinforcement learning* in the form of the question - "How can we guarantee online performance of a system during learning?" This implies an implementation of an adaptive mechanism for avoiding risk during learning, which is essentially the focus of this thesis.

1.2 Risk-Directed Exploration

Survival is a delicate balancing act. In an unpredictable and potentially hazardous environment, a single wrong choice of action may lead to fatal and irrecoverable consequences. The ability to assess the amount of immediate risk in any actions allows one to make minute-by-minute tradeoffs between attaining and abandoning a goal in order to ensure survival. In a hazardous environment, the question is not *what* to learn, but *whether* or not to learn (Kruusmaa, 1999).

During the course of learning, the agent makes decisions about which action to choose, either to find out more about the environment or to take one step closer towards the goal. In reinforcement learning, techniques for selecting actions during the learning phase are called exploration methods. Most exploration methods are based on heuristics, and rely on statistics collected from sampling the environment. Their goal is to sample the state space efficiently. All of these exploration methods are blind to the risk of actions.

In addressing the issue of self-preservation during learning, this thesis proposes a heuristic, model-based exploration method which selects actions

based on a measure of risk. This approach is different from previous work in that 1) a local, instead of global, measure of risk is used, and 2) the control of risk is done on a step-by-step basis during the learning phase. Our method also opens the possibility that the level of risk aversion of the agent can be customized or adjusted dynamically during the learning phase, although we do not explore this in detail in this thesis.

This thesis is organized as follows:

In Chapter 2, we provide an overview of the RL framework and methods for solving sequential decision problems. First, we introduce Markov Decision Processes (MDP) and reinforcement learning. We discuss various methods for solving MDPs. Finally, we explain the exploration-exploitation tradeoff, which arises naturally in the reinforcement learning framework, and review exploration methods that aim to address this tradeoff.

Chapter 3 introduces the notion of risk by drawing insights from decision theory in economics, which offers an argument as to why the measure of risk can be useful in the valuation of a prospect. We also review previous works on the control of risk in reinforcement learning.

Chapter 4 describes the model-based, risk-directed exploration method, which is the main contribution of this thesis. We provide a definition of risk, a justification for this definition, and an algorithm for incorporating risk sensitivity in the exploration process.

Chapter 5 presents experimental results demonstrating the behaviour of agents using this exploration technique in different environments.

Chapter 6 concludes the thesis by providing a review of the strengths and weaknesses of this approach, offering suggestions for future work.

CHAPTER 2
Sequential Decision Making in Uncertain Environments

In wisdom gathered over time, I have found that
every experience is a form of exploration.
- Ansel Adams

Decision making in uncertain environments is inherently risky. Actions often lead to many possible outcomes, some of which may have fatal consequences. The utility of an action towards a long term goal is unknown, and must be estimated by trial and error. Reinforcement Learning is a method for solving the sequential decision problems when a model of the environment is not known. Hence, the problem of risk arises naturally in this framework. This chapter provides a detailed review of reinforcement learning, the exploration-exploitation dilemma and existing solutions for addressing it.

2.1 Markov Decision Processes

Markov Decision Processes (MDP) (Bellman, 1957) are used to model sequential decision making. Formally, a finite MDP can be represented by the tuple $\{S, A, P, R\}$, where S is a discrete finite set of states in the environment, A is a discrete finite set of available or permissible actions within the environment, P is a matrix consisting of the probabilities of transitioning from state s to state s' given that action a is taken, specifically,

$$P^a_{ss'} = Pr(s_{t+1} = s' | s_t = s, a_t = a) \quad \forall t$$

and R is the matrix containing the expected reward associated with the transition, where

$$R^a_{ss'} = E[r_{t+1}|s_t = s, a_t = a, s_{t+1} = s'] \ \ \forall t$$

In this model, the state representation retains the relevant past history. Hence, the system has the Markov property - the next state and reward can be predicted given only the current state and action.

The goal of a decision problem is to find a way of choosing actions which maximizes a measure of performance. In this case, the measure of performance is the long term expected reward, i.e. $E[\sum_{t=0}^{T'} \gamma^t r_t]$, where T' is the number of decision epochs for a finite horizon problem. In an infinite horizon problem, $T' = \infty$ and γ is a discounting factor which bounds the sum by weighting less rewards that are received further in the future. A *policy* (Puterman, 1994) π is a strategy for selecting actions, where $\pi(s, a)$ is the probability of taking action a in state s under policy π (Sutton and Barto, 1998).

Markov Decision Processes provide a general framework for modeling sequential decisions. Their flexibility lies in the fact that actions, state transitions, and rewards can be either deterministic or stochastic, and hence a variety of environments can be represented. In addition, this representation lends itself well to divide-and-conquer solutions. Specifically, the problem of finding an optimal policy can be decomposed into subproblems of finding, for all states, the optimal utility value of the state s, or the optimal utility value of each action a taken at that state. The state utility value represents the expected return when starting in state s, following π, i.e.

$$V^\pi(s) = E_\pi[R_t|s_t = s] = E_\pi[\sum_{k=0}^{\infty} \gamma^k r_{t+k+1}|s_t = s]$$

where the return R_t is the sum of reward from time t to the end of the episode. Similarly, the action utility value represents the expected return of starting from state s, taking action a and following π thereafter, i.e.

$$Q^{\pi}(s,a) = E_{\pi}[R_t|s_t = s, a_t = a] = E_{\pi}[\sum_{k=0}^{\infty} \gamma^k r_{t+k+1}|s_t = s, a_t = a]$$

The optimal policy π^* is defined as a policy which has the best value at all states:

$$V^{\pi^*}(s) >= V^{\pi}(s) \forall s \forall \pi$$

The question that remains is how to compute $V^{\pi^*}(s)$ for each state. Bellman (Bellman, 1957) showed that the utility value of a state can be rewritten in terms of values of its successor states i.e.,

$$\begin{aligned}
V^{\pi}(s) &= E_{\pi}[R_t|s_t = s] \\
&= E_{\pi}[\sum_{k=0}^{\infty} \gamma^k r_{t+k+1}|s_t = s] \\
&= E_{\pi}[r_{t+1} + \gamma \sum_{k=0}^{\infty} \gamma^k (r_{t+k+2}|s_t = s)] \\
&= \sum_a \pi(s,a) \sum_{s'} P^a_{ss'}(R^a_{ss'} + \gamma E_{\pi}[\sum_{k=0}^{\infty} \gamma^k r_{t+k+2}|s_{t+1} = s']) \\
&= \sum_a \pi(s,a) \sum_{s'} P^a_{ss'}(R^a_{ss'} + \gamma V^{\pi}(s'))
\end{aligned} \tag{2.1}$$

Following the same rationale, the *optimal* utility value of a state or a state-action pair can be written as follows:

$$\begin{aligned}
V^{\pi^*}(s) &= \max_{a \in A_s} Q^{\pi^*}(s,a) \\
&= \max_{a \in A_s} \sum_{s'} P^a_{ss'}(R^a_{ss'} + \gamma V^{\pi^*}(s'))
\end{aligned} \tag{2.2}$$

$$Q^{\pi^*}(s,a) = \sum_{s'} P^a_{ss'}[R^a_{ss'} + \gamma \max_{a' \in A_{s'}} Q^{\pi^*}(s',a')]$$

where s' represents a possible next state given an action a is taken in state s. The Bellman optimality equations form a system of N equations, where N is the number of states. If the entire model is known, i.e. the P and R matrices are available, then the system of equations can be solved using dynamic programming. The result of this computation is the optimal value function, which attains $V^{\pi^*}(s)$ for each state or $Q^{\pi^*}(s,a)$ for each state-action pair. Given the optimal value function, the optimal behavior in an environment is to select at each state the action with the highest long term utility.

2.1.1 Methods for Solving MDPs

The recursive structure of the Bellman equation makes dynamic programming a suitable method for solving Markov Decision Processes, if the complete model is known. Dynamic programming relies on the fact that the optimal policy can be broken down and iteratively reconstructed from individual optimal policies of the subproblems involving the last stage of the computation, the last two stages, the last three stages, and so on, until the entire policy is constructed. This has been called a *backward induction method* (Puterman, 1994). The dynamic programming method, however, has a major drawback in that it cannot be applied directly when the model of the Markov Decision Process, i.e. the state transition matrix P and the reward matrix R, are unknown. Reinforcement learning provides algorithms for solving Markov Decision Processes when the model is not known.

One approach is to collect statistics from the environment in order to build a model of the MDP, and use this model to compute the optimal policy for the MDP. Specifically, if $n(s'|s,a)$ is the number of times state s' is encountered when action a is performed in state s, and $n(s,a)$ is

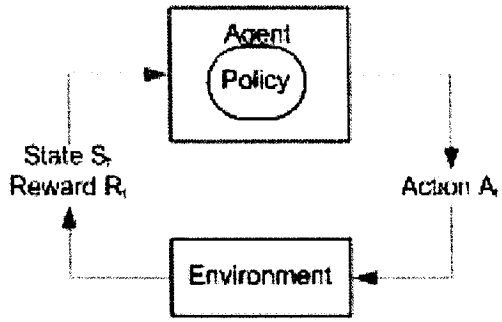


Figure 2–1: Reinforcement learning framework

the number of times action a is performed in state s, then the transition probabilities can be estimated as

$$\hat{P}^a_{ss'} = \frac{n(s'|s,a)}{n(s,a)}$$

Likewise, the reward for each transition can be estimated as

$$\hat{R}^a_{ss'} = \frac{\sum_{i=0}^{n(s'|s,a)} r_i}{n(s'|s,a)}$$

where r_i is the i^{th} sample of the reward observed for the particular transition.

Another approach is to learn the value function directly. A class of solutions known as temporal difference (TD) learning estimates the values of a state based on the immediate rewards and the estimated values of the next states, without using an explicit model of the environment. For example, TD(0), the simplest temporal difference method, updates the value estimate by moving it towards a new sample estimate as follows,

$$V_{t+1}(s) \leftarrow V_t(s) + \alpha[r_{t+1} + \gamma V_t(s_{t+1}) - V_t(s_t)]$$

Q-learning (Watkins and Dayan, 1992) and Sarsa (Sutton and Barto, 1998) are two examples of TD-based control algorithms. Instead of estimating the utility values of states, these methods use TD-style updates

to estimate the utility values of state-action pairs. At every time step, the agent selects an action based on the current state, observes the next state and the associated reward, and updates the action value function. In Q-learning, the action value is updated with the value estimate of the best possible action in the next state, even if that action is not taken in the next time step, i.e.

$$Q_{t+1}(s_t, a_t) = Q_t(s_t, a_t) + \alpha[r_{t+1} + \gamma \max_{a' \in A_{s_{t+1}}} Q_t(s_{t+1}, a') - Q_t(s_t, a_t)]$$

Because of this property, Q-learning is termed an *off-policy* algorithm and is essentially optimistic. Theoretical analysis proves that action values will converge to the optimal action-value function Q^{π^*} with probability 1 given that each action is executed infinitely often in each state, and if an appropriate schedule for decreasing the learning rate α is chosen (Watkins and Dayan, 1992).

In contrast, Sarsa performs the action-value update using an estimate based on the action that is actually taken in the next time step, i.e.

$$Q_{t+1}(s_t, a_t) = Q_t(s_t, a_t) + \alpha[r_{t+1} + \gamma Q_t(s_{t+1}, a_{t+1}) - Q_t(s_t, a_t)]$$

In other words, Sarsa is an *on-policy* method and results in more pessimistic behavior than Q-learning. Both Q-learning and Sarsa are considered *model-free* or direct methods because no explicit model of the environment is built during learning.

For both model-based and model-free methods, efficient exploration is an indispensible part of learning. In order to learn an accurate value function, the agent must try different actions to learn about their utility values. Simply adhering to doing what is known to be best, without

having explored the environment widely, usually results in behavior that is suboptimal.

2.2 The Exploration-Exploitation Dilemma

The study of the tradeoff between exploration and exploitation dates back to 1960 when Feldbaum (Feldbaum, 1960) first introduced dual control theory. The idea is that any controller, when placed in an environment with unknown parameters, has two conflicting goals - to perform control according to the best estimated parameters, i.e. *exploitation* of existing knowledge, or to probe the environment to derive more accurate parameters for better control, requiring *exploration* of unvisited states.

The exploration-exploitation problem is a central theme in the research on the k-armed bandit problem (Berry and Fristedt, 1985), which is the simplest possible reinforcement learning problem. The words *k-armed bandit* refer to a slot-machine with k arms. A decision maker is given an opportunity to have a fixed number of pulls on any of the k independent arms. There is a payoff generated according to an unknown underlying probability distribution unique for each arm. No cost is incurred except for the waste of a pull. Some arms are better than others in that they deliver higher expected reward.

Since knowledge of the statistics of each arm can only be obtained through trial and error, the decision maker must strike a balance between choosing the best arm according to his currently limited knowledge, and choosing another arm in order to verify or correct his current model of the machine. The question is how much the agent should explore, i.e. at which point he should stop probing the system and choose the best arm in order to maximize his reward. Some formal theories of exploration for the k-armed bandit problem include Gittins Allocation Indices (Gittins, 1998),

Learning Automata (Narendra and Thathachar, 1989) and the dynamic programming approach (Berry and Fristedt, 1985), the details of which will not be discussed here.

Solving MDPs using reinforcement learning is also in part the problem of optimizing the exploration-exploitation tradeoff. In Q-learning and Sarsa, for example, it is not advantageous for the agent to always select the best known action during the learning phase. The reason is that the agent can only select the best action based on what it knows, which is limited when the model is largely unknown at the beginning of learning. To improve its behavior, the agent should attempt other actions, even at the cost of incurring negative rewards, in order to acquire a more complete and accurate model. Exploration and exploitation are mutually dependent (Thrun, 1992b). In order for the cost of exploration to be minimized, exploitation is needed; in order for exploitation to reap the most benefit by taking place in the most relevant part of the state space, exploration is needed. Learning is most efficient when there is an optimal balance between exploration and exploitation such that the cost incurred is minimal and the reward gained in the long run is maximal.

2.2.1 Undirected verus Directed Exploration Methods

During learning, the agent must select actions whose consequences are unknown in order to learn about their effects. The action selection policy that decides exactly which of these unknown actions to explore is called an *exploration policy.*

The simplest strategy to steer from always picking the best known action is to select an action randomly at least sometimes. The action selection policy called ϵ-greedy, for example, selects the action with the best expected utility with probability $1 - \epsilon$ and selects actions uniformly

randomly with probability ϵ. Some variants of this strategy set ϵ to a large value at the beginning of the learning trial to encourage exploration, and decay ϵ over time.

The *softmax* action selection policy improves upon ϵ-greedy by selecting actions with probability weighted according to their action values, using a Boltzmann or Gibbs distribution. At any given time step t and state s, it chooses action a_t with probability

$$\pi(s_t, a_t) = \frac{e^{Q_t(s_t,a_t)/\tau}}{\sum_{b=1}^{|A_{s_t}|} e^{Q_t(s_t,b)/\tau}}$$

The *temperature* parameter τ controls whether actions are chosen greedily, as $\tau \longrightarrow 0$, or almost uniformly randomly, when $\tau \gg 0$.

These strategies are *undirected* in that they do not explicitly distinguish which states are worth visiting or which actions are worth performing. At the beginning of learning, undirected exploration essentially reduces to a random walk. Its complexity is proven, in the worst case, to be exponential in the number of steps needed to reach goal state (Whitehead 91). In the case of softmax, the strategy relies on expected utility as an accurate measure of the *goodness* of an action. This assumption can be problematic if the expected utilities of the competing actions are very close, or if the expected utility is still inaccurate due to insufficient sampling of the state space.

There has been considerable research on developing exploration methods that lead to efficient sampling of the state space. By *efficient*, it is meant that there should be sufficient, yet non-redundant, coverage of the state space such that an accurate estimate of the optimal value function is derived with as little computation as possible. Relying more on choice than chance, these so-called *directed* exploration methods exploit statistics

gathered in the environment to discriminate between unknown states, in order to determine their potential in revealing useful information. Besides the few theoretical approaches, e.g. E3 (Kearns and Singh, 2002) and R-Max (Brafman and Tennenholtz, 2002), most directed exploration methods, including the one presented in this thesis, are based on heuristics.

The central question in efficient exploration is - which states or actions are interesting, informative and thus worth exploring? The idea is that the probability of visiting (or revisiting) a state or state-action pair should be proportional to a heuristic measure indicating how *interesting* or *informative* they are. At any time step, an action is selected greedily to maximize the expected utility plus an *exploration bonus*, i.e. $Q(s,a) + \delta(s,a)$. What the *exploration bonus* represents ultimately depends on the answer to the central question posed, and varies among different lines of research. The recency-based approach, for example, suggests that states that are visited least recently (Sutton, 1990) are likely to contain useful information. Yet other approaches propose to visit states that are visited least frequently (Thrun, 1992a; Sato *et al.*, 1988), or to perform actions which have the greatest variation in their utility estimates (Thrun and Moller, 1992; Schmidhuber, 1991; Moore, 1990), or the highest variance in the possible outcomes (Meuleau and Bourgine, 1999; Kaelbling, 1993). We now present these ideas in more detail.

2.2.2 Exploration Statistics and Bonuses

In this section, we review the choice and computation of exploration bonuses for different classes of directed exploration methods.

Based on the rationale that the least visited states contain information that is yet unknown to the agent, the *counter-based* method (Thrun, 1992a; Sato *et al.*, 1988) keeps a count of how many times each state or state-action

pair has been tried, and selects an action which has been attempted the least frequently. In (Thrun, 1992a), for example,

$$\delta(s,a) = \frac{c(s)}{E[c|s,a]}$$

where $c(s)$ is the number of times a state has be visited, and

$$E[c|s,a] = \sum_{s'} P^a_{ss'} c(s')$$

is the expected value of the count over the possible next states s', given that an action a is executed in state s. In other words, given similar Q values, the counter-based exploration bonus induces a preference for the action that leads, on average, to a next state that is visited the least often. One drawback of the counter-based method (Thrun, 1992a) is that the number of visits to a state cannot always be assumed to present a measure of the accuracy of the model. The reason is that the difficulty of learning the value function is not homogeneous across the state space. In some regions of the state space, the values are easy to learn, so a few visits suffice, while other regions may require substantial revisiting. In addition, the counter-based method does not take into account the recency of events.

The *recency-based* method (Sutton, 1990), in contrast, is based on the premise that if a state has not been visited recently, there is more uncertainty about that state due to a possibly changing environment, and hence, the state should be revisited. The exploration bonus here is

$$\delta(s,a) = \epsilon\sqrt{n(s,a)}$$

where $n(s,a)$ is the number of time steps that have elapsed since action a was executed in state s.

Instead of equating the uncertainty of a state to the amount of encounters the agent has with that state, the *error-based* method (Thrun and Moller, 1992; Schmidhuber, 1991; Moore, 1990) selects actions that are associated with the greatest change in the utility estimation of $V(s)$ or $Q(s,a)$ observed in the past.

In *variance-based* methods, the exploration bonus is based on second-order statistics about the outcomes of a given action, such as the variance or standard deviation. This idea originated from the *Interval Estimation* (IE) method (Kaelbling, 1993) for the bandit problem. In IE, two statistics are collected for each action: w, which is the number of successes of taking the action, indicated by the receipt of a positive reward, and n, the number of times the action has been tried. From these statistics, a confidence interval of the success probability p_a of receiving a positive reward is computed for each action. The algorithm chooses the action that has the highest upper bound of the confidence interval of this success probability. The upper bounds are initially optimistic, and tighten as more experiences are gathered. The upper bound can remain high for two reasons. Either the action has an accurate utility estimate, or the action has not been tried often enough. Hence, the result is an exploration policy which exploits by choosing actions with the highest expected utility, or explores by choosing actions that have not been adaquately sampled. Building on IE, the *IEQL* algorithm presented in (Meuleau and Bourgine, 1999) uses an exploration bonus that describes the degree of local uncertainty. This exploration bonus is added to reward and backpropagated by TD learning to derive a global measure of the bonus. The reason given is that a global measure is able to handle certain environments that are normally misleading if a local measure is used. The same technique is used by (Sutton, 1990) in deriving a global

measure of recency. This dichotomy between local and global measures of exploration bonuses is described in further detail in (Wilson, 1996).

2.2.3 Model-Based Approaches

A different strand of research in exploration methods, including the new algorithm presented in this thesis, are model-based (Christiansen *et al.*, 1991; Dayan and Sejnowski, 1996; Dearden *et al.*, 1999). One advantage of the model-based approach is that more specific information about the transition probabilities and rewards can be used to distinguish between actions. For example, Ratitch et al. (Ratitch and Precup, 2003) propose a method for guiding exploration using a combination of two characteristics of MDPs - state transition entropy and forward controllability. State transition entropy (STE) measures the amount of stochasticity in the environment, i.e.

$$STE(s,a) = -\sum_{s' \in S} P^a_{ss'} \log P^a_{ss'}$$

and forward controllability (FC) measures how much the agent's actions actually impact the trajectories that the agent follows, and is computed from state transition entropy and its conditional form, i.e.

$$FC(s,a) = \sum_{s' \in S} P^a_{ss'} C(s')$$

where

$$C(s) = \frac{H(O_s) - H(O_s|A_s)}{H(O_s)}$$

$$H(O_s) = -\sum_{s' \in S} \left(\frac{\sum_{a \in A} P^a_{ss'}}{|A|}\right) \log\left(\frac{\sum_{a \in A} P^a_{ss'}}{|A|}\right)$$

$$H(O_s|A_s) = \sum_{a \in A} \frac{1}{|A|} \sum_{s' \in S} P^a_{ss'} \log P^a_{ss'}$$

where $O_s \in S$ is a random variable that represents the outcome of a uniformly random action in state s and A_s is an action chosen from a uniform distribution. The exploration bonus is, then, a weighted combination of

these two measures. In visiting actions with high state transition entropy and forward controllability, it is shown experimentally that more different states will be encountered, which consequently leads to a more homogenous exploration of the state space.

Some model-based exploration methods take the Bayesian approach (Wyatt, 2001; Dearden *et al.*, 1999). Bayeisan exploration (Dearden *et al.*, 1999) exploits ideas from information theory. Their algorithm estimates the benefit of exploring a state by assessing the agent's uncertainty about its current value estimates for that state. This is done by noting the difference between the value estimates produced by the current model and the *true* value estimate derived from a distribution of possible models. In this framework, the *value of information* of a state-action pair is high if the knowledge of the *true* value estimate changes the agent's policy in a significant way, i.e. when an action previously considered suboptimal is now considered the best choice, or when an action previously thought of as best is actually inferior. This method is computationally expensive, when compared to the online computation of exploration bonuses using a model, as proposed by (Ratitch and Precup, 2003).

To reiterate, learning a model online and using it to compute a bonus for guiding exploration has two major advantages. First, exploration bonuses are computationally cheap to compute. Second, the exploration bonus based on a model is a richer representation of the probabilistic charcteristics of actions, which in some cases, allows actions to be better distinguished. The model-based exploration method presented in this thesis is in part based on this rationale.

CHAPTER 3
Reflection on Risk

To make a mistake is only human; to persist in a mistake is idiotic.
- Cicero 106BC-43BC

During exploration, it is crucial for a policy to seek out informative states by selecting actions that lead to them. In an uncertain environment, however, the effects of actions are not known and can be potentially catastrophic. Despite the information they may reveal, actions may *not* always be worth taking. It is important to ask how much risk the agent should be allowed to tolerate for any particular piece of information. This chapter is intended to bring to light some intuition about the concept of risk and risk attitude, and to review the literature on risk in reinforcement learning.

3.1 Decisions under Risk

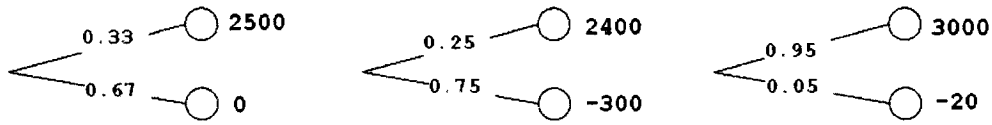
Decisions are said to be made under *uncertainty* or *risk* when there is ignorance about the data due to the lack of perfect or complete information (Taha, 1992). Despite their connections, *decisions under risk* and *decisions under uncertainty* are distinguished by economists as two different categories of decision making situations. In decisions under risk, an agent that is faced with a set of actions, whose effects are unknown but can be represented in terms of a probability distribution of outcomes. In decisions under uncertainty, no assumptions about the probability distribution of outcomes can be made.

In the context of decisions under risk, actions are essentially equivalent to lotteries, where a lottery l_i is a set of outcomes o_i, each of which occurs

with probability p_i and is associated with a specific reward r_i. Consider the example in Figure 3–1. How would an agent choose between the three lotteries? One way is to select the lottery l_i which has the highest expected reward value $EV(l_i)$

$$EV(l_i) = \sum_{i=1}^{n} p_i r_i$$

where n is the number of possible outcomes in the lottery. Under this criterion, the agent should choose lottery l_3.



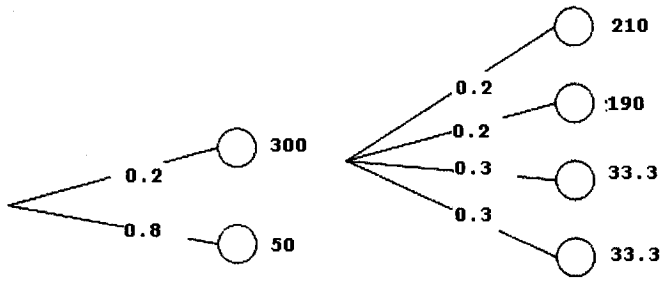
(a) Lottery l_1 (b) Lottery l_2 (c) Lottery l_3

Figure 3–1: Actions as lotteries

In this framework, there are many ways to define the *risk* of an action. Intuitively, risk implies the possibility of loss. Thus, risk can be defined as the probability of an event with negative consequences. In Figure 3–1, for example, lottery l_2 may be seen as the most risky because it has the highest probability of leading to the most negative outcome.

Risk is also related to the stochasticity of the outcome of an action. An action that leads to a sure gain has zero risk. In contrast, an action that leads to many possible outcomes has high risk because its effects are less predictable. Consider the example shown in Figure 3–2 where two actions have the same expected reward, but different distributions of stochastic outcomes. Action a_2 leads to four possible outcomes, two of which occur with probability 0.20 and give rewards of 210 and 190 respectively, and two others which occur with probability 0.30 and give a reward of 0.33. Action a_1 leads to two outcomes, a reward of 300 with probability 0.2 and a reward of 50 with probability 0.8. While the utility of the two actions are equal in

terms of expected value, a_1 is deemed less risky than a_2 because in addition to generally higher reward, there are fewer possible outcomes and thus the effects of the action are more predictable.



(a) Action a_1 (b) Action a_2

Figure 3–2: The stochasticity effect

These examples illustrate the fact that expected value is incapable of fully capturing and distinguishing the probabilistic characteristics of each action. Therefore, an alternative measure such as *risk* may be useful in the valuation of prospects in a decision.

3.2 Risk Attitude and the Avoidance of Danger

In 1738, Bernouilli introduced the famous St. Petersburg Paradox which suggests that the perceived value of a monetary reward is not necessarily equal to the amount of reward itself, but some other function of it. To explain the St. Petersburg Paradox, imagine the following bet where the player has to guess how many tosses of a coin are needed before it turns up heads. A player pays a fixed initial amount, and receives $\$2^n$ if the coin comes up heads on the n^{th} toss. The expected value of the gain is

$$\frac{1}{2}(2) + \frac{1}{4}(4) + \frac{1}{8}(8) + ... = 1 + 1 + 1... = \infty$$

Although the gain is infinite, there are few people who would be willing to pay a large upfront sum on this bet. This leads to the idea that the utility function is a subjective measure, which can mathematically represent

different individuals' preferences over lotteries. The rational behavior, under the maximum expected utility (MEU) principle (Neuneier and Mihatsch, 2002), is to select the lottery l_i which has the highest expected *utility* value $EU(l_i)$,

$$EU(l_i) = \sum_{i=1}^{n} p_i u(r_i)$$

According to the MEU principle, a person is risk-neutral if the utility function is linear, i.e. $u''(r) = 0$, risk-averse if the utility function is concave, i.e. $u''(r) < 0$, and risk-seeking if the utility function is convex, i.e. $u''(r) > 0$. These utility functions are depicted in Figure 3–3.

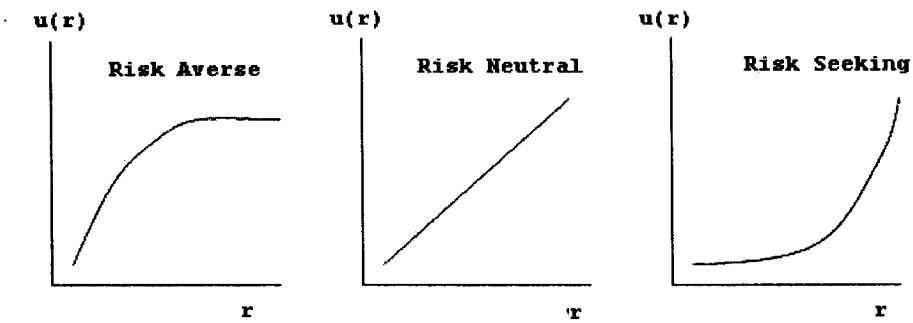


Figure 3–3: The shape of the utility function and risk attitude

The risk attitude of an agent can be explained by how much the agent is willing to pay for a certain gamble. If a certain lottery yields expected utility q, then a fair bet would be an amount that is equal to q. In other words, the agent should be indifferent towards the lottery which yields an expected amount of q and having q for certain. This amount q is called the *certainty equivalent* of the lottery.

According to the Jensen Inequality, a concave function has the property $u(E[r]) <= E[u(r)])$. This suggests that due to diminishing return, the utility of the expected utility is in fact less than the expected utility itself.

An agent with this utility function will be less willing to pay the amount q in the gamble. In other words, the certainty equivalent of the lottery is less than the expected utility. The agent is considered risk-averse. A comparable argument can be made for using a convex function to characterize risk-seeking attitude.

In control theory, some studies have attempted to transform the cumulative return by a non-linear utility function and optimize policies based on this transformed criterion (Howard and Matheson, 1972; Hernandez-Hernandez and Marcus, 1966; Koenig and Simmons, 1994). A prevalent choice of non-linear utility function is the exponential function, since it has certain properties that make the problem feasible to solve using dynamic programming (Pratt, 1964; Howard and Matheson, 1972). However, in general, most non-linear utility functions violate Jensen inequality, specifically,

$$u(\sum_{t=0}^{\infty} \gamma^t r_t) \neq \sum_{t=0}^{\infty} \gamma^t u(r_t)$$

in a infinite horizon sequential decision problem. As a result, the use of subjective utility does not lend itself easily to dynamic programming or model-free reinforcement learning methods such as TD(0) or Q-learning (Heger, 1994).

3.3 Consideration of Risk in Reinforcement Learning

The attitude of the agent towards risk can be exploited for the purpose of avoiding danger in other ways. In some domains where the safety of the agent is particularly important, for example in robotics where equipments are expensive, some researchers have added to the control system explicit reflex behavior (Milan, 1996) or domain knowledge (Singh *et al.*, 1994). In (Singh *et al.*, 1994), actions are no longer primitive but themselves closed-loop policies for avoiding collision. Experimental results (Singh *et al.*, 1994)

for two navigation tasks show that learning with actions that have no built-in domain knowledge took 2000 times more trials to reach the same level of performance and crashed into walls more than 200 times. This type of formulation, evidently, requires the use of strict conditions and very specific domain knowledge, which may not be possible or available in every scenario. The use of prior knowledge also diminishes the autonomy of the agent.

Risk-averse behavior can be induced by many other methods, e.g. by solving an objective function that is penalized by the variance of the return (Sato and Kobayashi, 2000), by updating the value function based on a pessimistic estimate (Heger, 1994; Gaskett, 2003), minimizing the probability of entering fatal states (Geibel, 2001), or transforming the temporal differences to more heavily weight events that are unexpectedly bad (Neuneier and Mihatsch, 2002). All of the above methods use direct learning. Although there is an underlying probability distribution for the MDP, the transition probabilities and rewards are not explictly available. Therefore, these methods fall under the category of *decision under uncertainty.* Risk here refers to cost or, in an implicit sense, the possibility of the occurrence of negative events.

3.3.1 Transforming Temporal Difference

Neuneier and Mihatsch (Neuneier and Mihatsch, 2002) proposes a risk-sensitive control framework that shares the same limiting behavior as the exponential utility approach, but which is also adequate for learning. Their approach is to transform the temporal differences by overweighing transitions to sucessor states where the immediate return happens to be smaller than in the average, and underweighing transitions to sucessor states where the immediate return happens to be larger than in the average.

$$Q_{t+1}(s_t, a_t) = Q_t(s_t, a_t) + \alpha\chi^{\kappa}[r_{t+1} + \gamma \max_{a' \in A_{s_{t+1}}} Q_t(s_{t+1}, a') - Q_t(s_t, a_t)]$$

where χ^{κ} is a weighting function such that

$$\chi^{\kappa} : x \mapsto \begin{cases} (1-\kappa)x & \text{if } x > 0 \\ (1+\kappa)x & \text{otherwise.} \end{cases}$$

and $\kappa \in (-1, 1)$ is a scalar parameter for specifying the desired risk attitude. When $\kappa = 0$, the update rule is the same as the standard Q-learning update rule, and the policy is risk neutral . When $\kappa \to 1$, the policy is risk-averse. The reason can be illustrated by the following example. If $r_{t+1} > 0$ and the action values of the current state $Q_t(s_t, a_t)$ and next state $Q_t(s_{t+1}, a_{t+1})$ are both positive, then this update rule will underweigh r_{t+1} and the $\max_{a' \in A_s} Q_t(s_{t+1}, a')$, and overweigh the $Q_t(s_t, a_t)$ term. On the other hand, if $r_{t+1} < 0$ and the action values for both states are both negative, the update rule will overweigh r_{t+1} and the $\max_{a' \in A_s} Q_t(s_{t+1}, a_{t+1})$, and underweigh the $Q_t(s_t, a_t)$ term. This is equivalent to saying that the agent is pessimistic about the positive rewards, and at the same time, over-emphasizes the negative rewards received in the next state. Under similar logic, the policy is risk-seeking when $\kappa \to -1$.

The risk parameter κ needs to be chosen carefully in order to facilitate learning in any given environment. One suggestion (Neuneier and Mihatsch, 2002) is to set κ to a small value until the algorithm converges, and then increase κ for subsequent runs. This suggests that the agent should be risk-seeking at the beginning of learning in order to gather information, and then become increasingly risk-averse as the amount of information increases. In allowing for risk-averse and risk-seeking attitude at different times during

learning, the agent is given the flexibility to exhibit a range of behavior depending on the current context.

3.3.2 Risk as Variance

In finance, Markowitz portfolio theory (Markowitz, 1952) suggests that the risk of individual investments can be measured by considering the deviation of individual investments from the mean of the portfolio return, i.e. variance. Several works in control theory study the use of mean-variance analysis to solve this problem and attempt to maximize the variance-penalized reward (Filar *et al.*, 1989; Huang and Kallenberg, 1994; Sobel, 1982; White, 1992; White, 1994). An example of the *expected value-variance criterion* is given by (Taha, 1992) as

$$\max E[R] - \lambda var[R]$$

where R is the cumulative return, and λ is a pre-specified constant known as the *risk aversion factor*, which indicates how much variance is weighted. Weighting the variance by a large λ implies that the agent is sensitive to large reductions in reward below $E[R]$ (Taha, 1992). An equivalent formulation for TD learning is provided by (Sato and Kobayashi, 2000).

There are limitations to the approach to use variance as a measure of risk. First, the fat tails of the distribution are not accounted for. Consequently, risk can be underestimated due to the ignorance of low probability, but highly severe events. Second, variance penalizes both positive and negative risk equally and does not distinguish between the two. Finally, this measure is not consistent with the expected utility approach, unless returns are normally distributed or the utility function is quadratic. The Markowitz model of risk has been "incorrectly applied to many cases in which risk cannot be described by variance, dependence cannot be measured

by linear correlation coefficient, and utility function does not even dream to be quadratic" (Szego, 2004).

3.3.3 Bounding Risk by Pessimism

Intuitively, risk can be minimized during learning if the agent is completely pessimistic about the outcomes of the actions. Heger (Heger, 1994) presented a variant of Q-learning called $\hat{Q}$-learning which follows the so called *maximin* criterion, under which action values are updated with the best of the worst outcomes of the next state, i.e.

$$Q_{t+1}(s_t, a_t) = \max[Q_t(s_t, a_t), r_{t+1} + \gamma \min_{a' \in A_{s_{t+1}}} Q_t(s_{t+1}, a')]$$

The $\hat{Q}$ value is essentially a lower bound on value. The policy that is learned is risk-averse and can be considered optimal under the assumption that the minimax criterion is accepted as a valid basis for rationality. This criterion is called *minimax* when the action value refers to cost instead of gain. The generally lower action values mean that the agent will see most states as worse than they really are and act in a risk-averse way.

$\hat{Q}$-learning and the minimax criterion are useful when the avoidance of risk is imperative. However, Gaskett tested $\hat{Q}$-learning in a stochastic cliff world environment, under the condition that actions are picked greedily, and found that $\hat{Q}$-learning demonstrated extreme pessimism which can be more injurious than beneficial. For example, the agent learns to jump off the cliff from the start square to avoid the higher cost of taking a few steps before the cliff fall accident occurs (Gaskett, 2003). Under a different condition, where actions are selected ϵ-greedily, $\hat{Q}$-learning found a risky path which follows very close to the edge of the cliff. Gaskett's criticism of $\hat{Q}$-learning is that the pessimistic behavior derived from the minimax criterion is suitable only in adversial games, but inappropriate for other problems. In general,

the minimax criterion is too restrictive as it takes into account severe but extremely rare events which may never occur (Neuneier and Mihatsch, 2002).

To avoid extreme pessimism, α-Hurwicz criterion provides a way for interpolating between extreme pessimism and extreme optimism using a weighting parameter $\alpha \in [0, 1]$ The criterion is given in (Taha, 1992) as

$$\max_{a_i}[\alpha \max_{s_j} Q(s_j, a_i) + (1 - \alpha) \min_{s_j} Q(s_j, a_i)]$$

The most optimistic behavior is produced when $\alpha = 1$ since actions are chosen according to $\max_{a_i} \max_{s_j} Q(s_j, a_i)$. In contrast, the most pessimistic condition is produced when $\alpha = 0$ since actions are chosen according to $\max_{a_i} \min_{s_j} Q(s_j, a_i)$, which is equivalent to the maximin criterion. In other words, the maximin criterion is a special case of Hurwicz α-criterion where $\alpha = 0$. A range of behavior moderated between optimishm and pessimism can be produced by the intermediate α values.

β-pessimistic Q-learning (Gaskett, 2003) is based on this criterion, where the action values are updated as follows,

$$Q_{t+1}(s_t, a_t) = Q_t(s_t, a_t)+\alpha[r_{t+1}+\gamma((1-\beta) \max_{a' \in A_{s_{t+1}}} Q_t(s_{t+1}, a')+\beta \min_{a' \in A_{s_{t+1}}} Q_t(s_{t+1}, a'))]$$

Note that setting β to 0 or 1 renders the equation into the standard Q-learning or the minimax algorithm respectively. Experimental results show that when $\beta = 0.5$, the algorithm reaches the same level of pessimism as $\hat{Q}$-learning, although the agent manages to reach the goal state in some cases, unlike in $\hat{Q}$-learning.

3.3.4 Risk as Probability of Fatal Event

Geibel (Geibel, 2001) defines risk by equating it to the probability of entering a fatal state, where a fatal state refers to a terminal state

that marks the end of a learning epsiode. This definition arises from the recognition that not only the magnitude, but also the probability of extreme events with high negative cost needs to be bounded. The proposed algorithm aims to find an optimal policy under which this probability is smaller than some threshold value. In particular, $\rho^{\pi}(s)$, the probability of the agent ending up in a fatal state when starting in state s and following policy π, should be bounded by $\omega \in [0, 1]$. States with the property of $\rho^{\pi}(s) < \omega$ are called *safe* states. A risk-minimal policy is one that possesses the maximum set of safe states possible. The algorithm solves a constrained MDP, deriving at the same time a value maximal policy and a risk minimal policy.

$\rho^{\pi}(s)$ is in fact not a probability in the standard sense, but defined (Geibel, 2001) as the expected value of the accumulated cost

$$\rho^{\pi}(s) = E \sum_{i=0}^{\infty} \bar{r}_i$$

where $\bar{r}$ is a cost indicator function that equals 1 when a fatal state is entered, and 0 otherwise. Since the probability of entering a fatal state depend on action, the corresponding probability of entering a fatal state for a state-action pair is defined as

$$\sigma^{\pi}(s, a) = \sum_{s'} P^{a}_{ss'} \rho^{\pi}(s')$$

At a given time step, the algorithm updates $Q_{t+1}(s_t, a_t)$ and $\sigma^{\pi}_{t+1}(s_t, a_t)$ and computes the σ-penalized action value

$$U^{\lambda}_{t+1}(s_t, a_t) = \lambda Q_{t+1}(s_t, a_t) - \sigma^{\pi}_{t+1}(s_t, a_t)$$

Since actions with different action and σ values can have the same U^{λ} value, actions are chosen according to the (1,2)-lexicographical ordering

of $(U^{\lambda}_{t+1}(s,a), Q_{t+1}(s,a), \sigma_{t+1}(s,a))$. Under this ordering, an action a_1 in a given state s is preferred over another action a_2 if $U^{\lambda}(s,a_1) \geq U^{\lambda}(s,a_2)$ and if this holds, $Q(s,a_1) > Q(s,a_2)$.

In the beginning, λ is set to 0 so that a near risk-minimal policy can be learned. Subsequently, λ is increased by ϵ until the number of safe states begins to decrease. Under this scheme, optimal Q-values can incrementally exert more influence without compromising the number of safe states.
In contrast with the strategy proposed by Neuneier and Mihatsch, this approach suggests that the agent should be risk adverse from the onset, i.e. until a maximum number of safe states is reached, and become risk-seeking thereafter.

CHAPTER 4
Directed-Exploration using a Risk Measure

True genius resides in the capacity for evaluation of uncertain, hazardous, and conflicting information.
- Winston Churchill (1874-1965)

4.1 Intuition

Common to many of the works reviewed in the previous section is the fact that risk-averse behavior is induced by transforming the action values. There are several reasons why this may not be desirable. First, if the action values are updated based on a conservative criterion, the policy may be overly pessimistic. Second, the worst thing that can happen to an agent in an environment may have high utility in the long term, but fatal consequences in the short term. Attention should be paid to both the short term consequences and long term utilities of actions. Third, the distortion of the action values means that the true long term utility of the actions are lost. Ideally, we would like an exploration method to react to immediate risk by manifesting different risk attitudes, while leaving the action values untouched.

With a model-based approach, the problem of *decisions under uncertainty* turns into *decisions under risk*, as the probabilities and rewards of the action outcomes are available. As seen in Chapter 3, different measures of immediate risk can be derived from these statistics, and computing an exploration bonus online is also computationally cheap.

Motivated by the above reasons, this thesis presents a new model-based directed exploration method that selects actions using an exploration bonus

that is based on risk. In this section, we will present the algoirthm in detail and an analysis about how the algorithm is expected to behave under different conditions.

4.2 The Risk Measure

A risky action generally implies that the action may lead to a negative event, or that the effects of the action are uncertain. The risk measure in this algorithm, which is a variant based on the definition proposed by (Yang and Qiu, 2005), incorporates these two intuitions that characterize risk.

Risk Measure 4.2.1 *Given a state, the measure of risk for a particular action is the weighted sum of the entropy and normalized expected reward of that action. This definition is adopted from (Yang and Qiu, 2005).*

$Risk(s,a) = \lambda H_a(s) - (1-\lambda)\frac{E[R^a_{ss'}]}{\max_{a \in A_s} |E[R^a_{ss'}]|}$

where

$H_a(s) = -P^a_{ss'} \log P^a_{ss'}$

$E[R^a_{ss'}] = \sum_{s'} P^a_{ss'} R^a_{ss'}$

The definition consists of an entropy term, describing the stochasticity of the outcomes of a given action in a given state, and a normalized expected reward term, describing the relative negativity of the possible outcomes of that action. These two terms are weighed with the parameter λ.

4.3 Algorithm

The risk measure of an action is combined with the action value to form the *risk-adjusted utility* of an action, i.e.

$$U_r(s,a) = p * (1 - Risk(s,a)) + (1-p) * Q(s,a)$$

where $p \in [0,1]$. The first term measures the safety value of an action, while the second term measures the long term utility of that action. The

parameter p, which we will call from now on the p-value, provides a way to interpolate between paying attention to the long term utility of an action, when $p \rightarrow 0$, and paying attention to safety, when $p \rightarrow 1$. In this thesis, the p-value is fixed at a pre-defined level.

To adjust the probability of an action being selected based on its riskiness, the *risk-adjusted utility* of an action is then substituted into the Boltzmann function instead of the Q-values, i.e.

$$\pi(s,a) = \frac{e^{\frac{U_r(s,a)}{\tau}}}{\sum_{b=1}^{n} e^{\frac{U_r(s,b)}{\tau}}}$$

As the p-value increases, the Boltzmann action selection rule selects the action with higher risk with exceedingly lower probability. In other words, the p-value controls the relative risk aversion of the agent.

In the Sarsa (Figure 4–1) and Q-learning framework, the risk value of each action is computed from a model of the MDP, which is either given or learned online. The risk and action value of the action is then used to construct the risk-adjusted utility which is used in the Boltzmann function to produce the probability of selecting that action. For Q-learning, the algorithm is the same except that the update rule is different.

Initialize $Q(s,a) = 0, n(s) = 0, n(s,a) = 0 \ \ \forall s \ \ \forall a$
Repeat (for each episode):
  Initialize s_t; $n(s_t) \leftarrow n(s_t) + 1$
  Choose a_t from s_t using Boltzmann with $U_r(s_t, a_t)$
  Repeat (for each step of episode):
    Execute action a_t, observe r_{t+1} and s_{t+1}; $n(s_t, a_t) \leftarrow n(s_t, a_t) + 1$
    Choose a_{t+1} from s_{t+1} using Boltzmann with $U_r(s_{t+1}, a_{t+1})$
    $Q_{t+1}(s_t, a_t) \leftarrow Q_t(s_t, a_t) + \alpha[r_{t+1} + \gamma Q_t(s_{t+1}, a_{t+1}) - Q_t(s_t, a_t)]$
    Update $Risk(s_t, a_t)$ and $U_r(s_t, a_t)$ using the collected statistics
    $s_t \leftarrow s_{t+1}$; $a_t \leftarrow a_{t+1}$
  Until s_t is terminal

Figure 4–1: Risk-directed exploration in Sarsa

4.4 Analysis

4.4.1 Effects of Risk on Probability of Action Selection

One question is how the riskiness of an action affects its probability of being picked. The answer can be established experimentally by considering the following hypothetical state with two actions, a_1 and a_2. The action value of a_2 is 0, while the action value of a_1 is varied from -1 to 1. The figure below shows the probability, computed by the risk-adjusted Boltzmann function, of a_1 being selected as the action value varies. This plot is done for the standard case where only the action value is used in the Boltzmann distribution, or where *risk-adjusted utilities* with various p-values, i.e. $p = 0.0$, $p = 0.2$, $p = 0.4$, $p = 0.6$, $p = 0.8$, $p = 1.0$, are used. In this experiment, $Risk(a_1) = Risk(a_2)$ and the temperature parameter is 0.05. Note that $p = 0.0$ is equivalent to the standard Boltzmann, and hence the curves for both are overlapping.

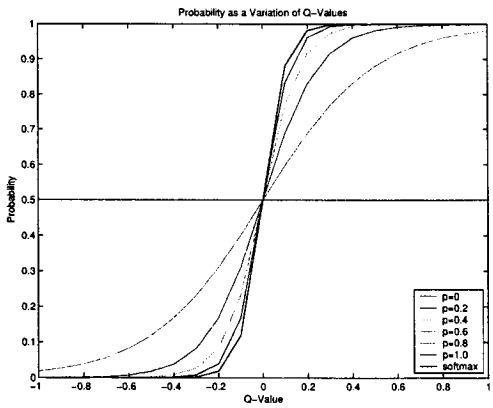


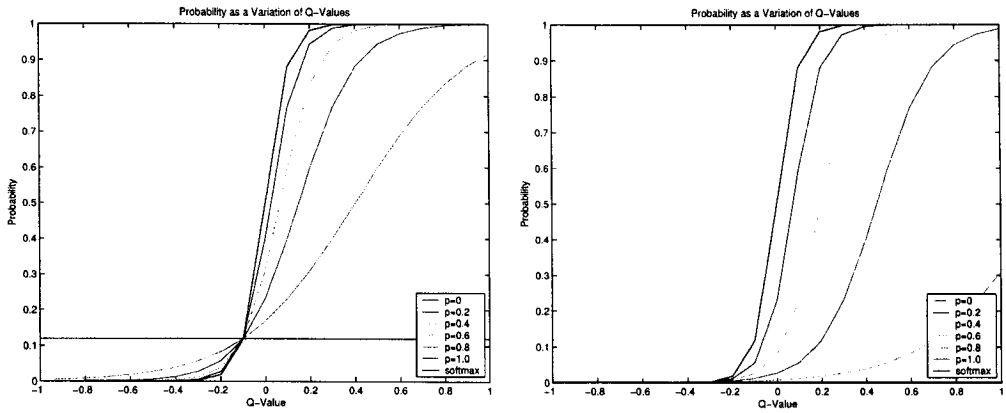
Figure 4–2: Different trends in the probability of selecting action a_1 for risk-directed exploration with different p-values

Figure 4–2 shows that for the standard Boltzmann exploration, a_1 has increasingly higher probability of being selected when its action value surpasses that of a_2 at 0, and lower probability of being selected when its

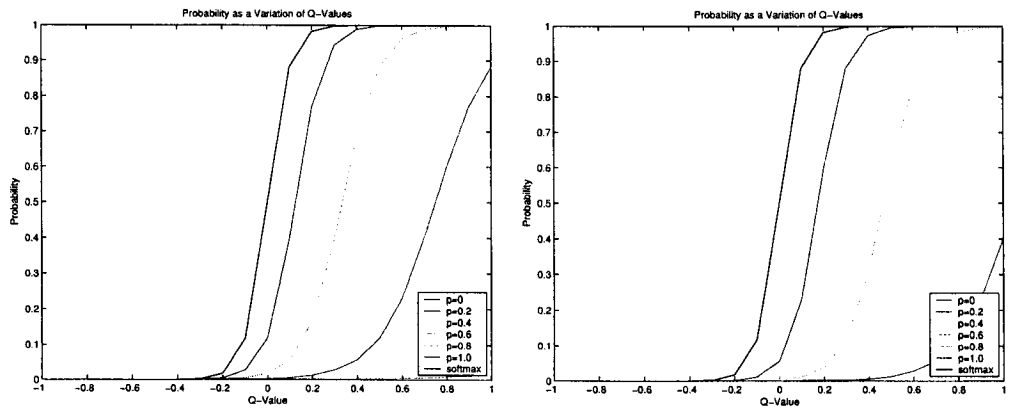
action value falls below 0. The probability curve is sigmoid-shaped. A similar trend is observed for the Boltzmann probability that uses risk-adjusted utilities, although the curves become flatter as the p-value increases, i.e. as the risk term is increasingly overweighted. At $p = 1.0$, the risk measure completely dominates the *risk-adjusted utility* value. Since the two actions have the same risk, they are picked with equal probability at all times.

For the intermediate p-values, there are two observations. As the p-value increases, the probability of a_1 is lowered. A greater difference in the predicted value is required in order for a_1 to be preferred. This is analogous to risk-averse behavior. However, it is also true that unless its value is very bad, a_1 still has some probability of being chosen. This is due to the fact that the *safety* term in the risk-adjusted utilities can be positive even for risky actions. As a result, it can potentially raise the risk-adjusted utilities of both good and bad actions, causing the bad actions to be selected more often than desirable. Some suggestions for transforming the Boltzmann probability function are provided in the conclusion of this thesis as future work.

What if a_1 has a different value of risk than a_2? The probability of selecting a_1 when the risk value of a_1 is higher than a_2 by 0.1, 0.3, 0.5, 0.7 are plotted in Figure 4–3. The value of a_2 is 0 at all times, while the value of a_1 is varied from -1 to 1. Within each plot in 4–3, the general trend induced by intermediate p-values is still observed, i.e. the higher the p-values, the flatter the curve. In addition, the more the risk value of a_1 increases, the higher the predicted value has to be in order for a_1 to be selected. Furthermore, the greater the p-value, the more drastically the action selection probability is depressed as the difference of the risk values between a_1 and a_2 becomes larger.



(a) $Risk(a_1) - Risk(a_2) = 0.1$ **(b)** $Risk(a_1) - Risk(a_2) = 0.3$

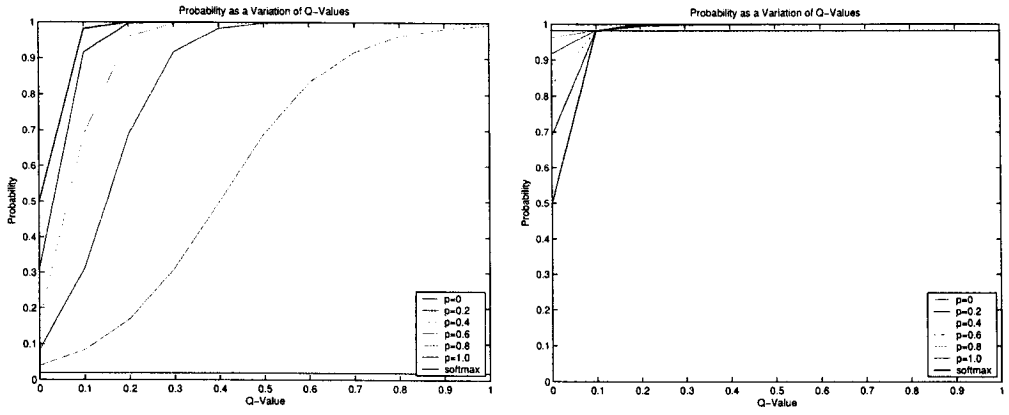


(c) $Risk(a_1) - Risk(a_2) = 0.5$ **(d)** $Risk(a_1) - Risk(a_2) = 0.7$

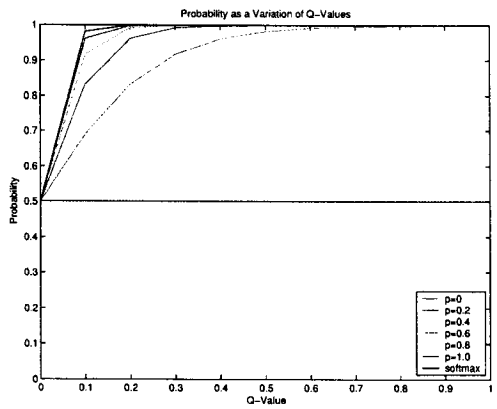
Figure 4–3: The probability of selecting action a_1 when the risk values of the two actions differ by varying amounts

A second, more realistic, scenario is that a_1 and a_2 are both initialized with value equal to 0. The value of a_1, the good action, approches 1 while that of a_2, the bad action, approaches -1. Figure 4–4 shows how the probability of selecting a_1 varies. The figure below shows three scenarios where Risk(a_1)=0.6 and Risk(a_2)=0.4 (Figure 4–4(a)), Risk(a_1)=0.4 and Risk(a_2)=0.6 (Figure 4–4(b)), Risk(a_1)=Risk(a_2)=0.5 (Figure 4–4(c)).

When a_1 has higher risk than a_2, its probability of being selected is lowered using the risk-adjusted Boltzmann function. On the other hand, when a_1 has lower risk than a_2, depending on the level of risk aversion induced by the intermediate p-values, its probability of being selected is high



(a) $Risk(a_1) > Risk(a_2)$ (b) $Risk(a_1) < Risk(a_2)$



(c) $Risk(a_1) = Risk(a_2)$

Figure 4–4: The probability of selecting action a_1 for two actions with diverging action values under different risk conditions

even when $Q(a_1) = Q(a_2) = 0$. This means that the using risk-adjusted utility, the Boltzmann action selection rule would always select the action with lower risk with much higher probability.

4.4.2 Effects of Varying λ

What is the contribution of entropy and reward in the various situations? In order to test the effects of varying λ, we test the performance of the algorithm under conditions where only the entropy term or the normalized expected reward term dominates the risk measure, or the two terms exert equal influence in the equation. In order to do that, we must set up an environment where the entropy is not uniform throughout the environment.

This may include the presence of obstacles, or local "slippery" regions. One such environment is shown in figure 4–5. In this environment, actions lead to deterministic outcomes everywhere except for the cells marked white. In this experiment, a fixed model is assumed.

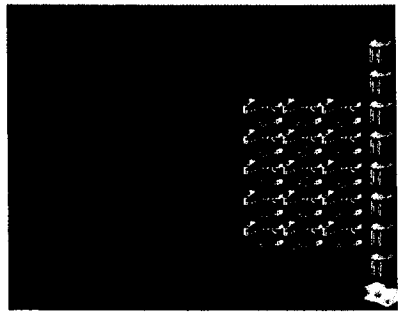
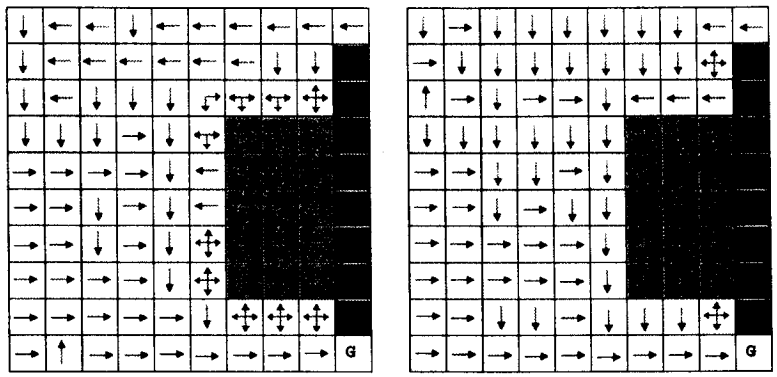


Figure 4–5: Slippery world

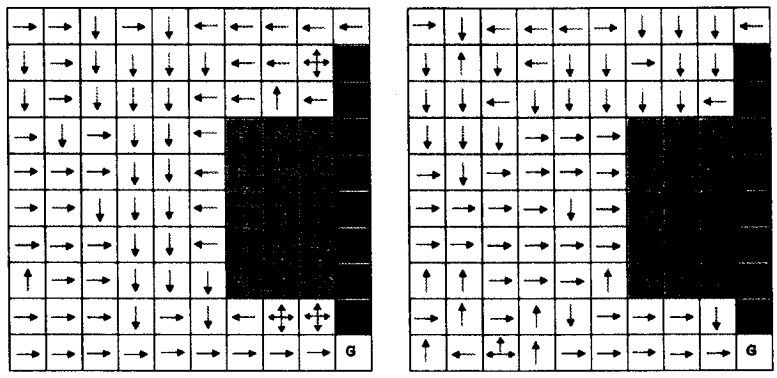
Figure 4–6 shows the policy learned using $\lambda = 1.0$ (Figure 4–6(a)), where the entropy term completely dominates the risk measure, and $\lambda = 0.0$ (Figure 4–6(b)), where the normalized expected reward dominates. The cliff is marked by red cells, the slippery regions by blue cells, the start state by the yellow cell and the goal state by the letter *G*.

At $\lambda = 1.0$, when the risk measure is represented by entropy only, the learned policy prefers to stay far away from the slippery region. In contrast, at $\lambda = 0.0$, when the risk measure is represented by normalized expected reward only, the learned policy yields a path around the slippery region that is much closer than that for $\lambda = 1.0$. The policy for the slippery region does not distinguish between the four actions, probably because the slippery region is not visited at all during the learning phase.

At $\lambda = 0.5$ (Figure 4–6(c)), the policy learned yields a path that is directed away from the cliff and the slippery region, similar to $\lambda = 1.0$. It can be observed, however, that the risk aversion is more tentative for



(a) $\lambda = 1$ (b) $\lambda = 0$



(c) $\lambda = 0.5$ (d) Boltzmann

Figure 4–6: Policy learned using Boltzmann with action values versus risk-adjusted utilities of varying λ

$\lambda = 0.5$. This is possibly because part of the slippery region and the surrounding area has been visited during learning, and hence a definite action is learned for those states. In constrast, for $\lambda = 1.0$ and $\lambda = 0.0$, some states have been avoided entirely during learning, as a result, many actions share equal utility and are not distinguished.

In contrast, the standard Boltzmann action selection method (Figure 4–6(d)) results in a policy that follows the shortest path to the goal, less concerned with the risk associated with the slippery region and the cliff. The preferred path seems to stay away from the cliff by one column, but straight through the slippery region.

4.5 Mult-Step Risk

Multi-step risk measure, e.g. two-step risk measure $Risk^{II}(s,a)$, can be used to evaluate an action, i.e.

$$Risk^{II}(s,a) = \lambda H_a^{II}(s) - (1-\lambda)\frac{E^{II}[R^a_{ss'}]}{\max_{a\in A}|E^{II}[R^a_{ss'}]|}$$

where

$$H_a^{II}(s) = H_a(s) + \sum_{s'} P^{a^*}_{ss'} H_{a^*}(s')$$

$$E^{II}[R^a_{ss'}] = \frac{\sum_{s'} P^a_{ss'}[R^a_{ss'} + \sum_{s''} P^{a^*}_{s's''} R^{a^*}_{s's''}]}{\max_a E^{II}[R^a_{ss'}]}$$

where a^* is chosen to maximize $Q(s',a')$.

By evaluating an action based on its riskiness over two steps instead of one, the agent is given the advantage to looking ahead in order to avoid states so close to the fatal states that slight stochasticity in the environment may get the agent there.

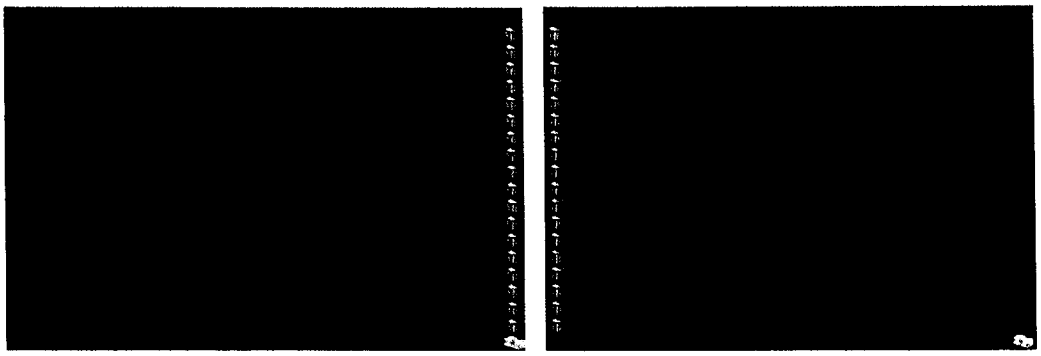
CHAPTER 5
Experiments

The risk-directed exploration method provides a framework in which the probability of an action being selected is adjusted depending on a weighted combination of its riskiness and utility. This chapter presents the results of experiments that investigate four major questions: (1) How do different weighted combinations of riskiness and utility, determined by the p-value, affect learning performance and the quality of the learned policy? (2) Is the performance of the algorithm somewhat preserved when the model is learned online? (3) How does the performance of the algorithm vary in different environments? (4) How does the performance of the risk-directed exploration method compare to that of other directed exploration methods, specifically the recency-based and counter-based methods?

5.1 Environments and Parameter Settings

An environment typically used in reinforcement learning to evaluate the sensitivity of algorithms to risk is the cliff world. In this environment, the objective of the agent is to travel from the start to the goal state without falling off the cliff. The *close-by-cliff* world and *far-away-cliff* world are two examples of such environments (Figure 5–1), and will be used in the experiments.

Different configurations of the cliffs in the environment render some directed exploration methods better than others. In environments where the goal can be easily reached without visiting the risky regions at all, the risk-directed exploration method has a obvious advantage over the recency-based and counter based methods. The recency-based method tends to direct the



(a) Close-by-cliff world **(b)** Far-away-cliff world

Figure 5–1: Environments

agent to less recently visited parts of the state space. In the case of the *far-away-cliff* world, this is where the cliffs are located. This cost is incurred unnecessarily, since the goal is within a short reach from the agent's starting position along a clear path. Using risk-directed exploration, however, the agent should learn to avoid the far away cliff region after only a few visits there. The conjecture that the risk-directed exploration method outperforms the recency-based and counter-based methods in these environments will be investigated experimentally.

In both worlds, the set of states is represented by the possible coordinates of the agent in terms of the row and column where the agent may be located. In a grid world with 20 by 20 tiles, the total number of states is 400. The terminal states include the location where the *goal* or the cheese is found, and the location of the cliffs. The agents are allowed four actions, i.e. A={up, left, right and down}. However, due to the constraints of the boundaries of the grid, the set of permissible actions A_s in each state may be smaller than A. With probability 0.8 that the agent will enter a state as intended, and with probability 0.2 it will slip into the neighboring cells of the intended destination. Finally, the reward for reaching the goal is +1, the penalties for falling off a cliff -1, and the reward for all other states is 0.

In our experiments, the *performance* of the algorithm is characterized by six measures: (a) the training score in terms of cumulative reward averaged (b) the testing score in terms of cumulative reward averaged (c) % of termination by cliff fall during learning (d) % of termination by reaching the goal during learning (e) the life span during training, in terms of the number of time steps elapsed until termination (f) in the case where the model is learned online, the amount of model error during training, measured by the averaged L1 distance between the learned and true transition probabilities and rewards for each state-action pairs in the model. The experiment is run over 100 episodes and all results are averaged over 20 runs. Optimized for performance, the parameters $\alpha = 0.25$, $\lambda = 0.5$ (except for the experiment investigating the effects of varying λ), and $\tau = 0.05$ are used for all experiments. The constants used in the counter-based and recency-based methods are both 400. In all experiments, the two-step risk measure is used. It has been shown in (Law *et al.*, 2005) that risk-directed exploration using the two-step risk measure produces results that are comparable to $TD(\lambda)$ where $\lambda = 0.7$, in terms of the percentage of cliff fall during training. Furthermore, evaluating actions using the two-step risk measure provides the agent with lookahead, which further accentuates the effects of risk aversion.

5.2 Results

In the analysis of the results, we compare the performance of the algorithm under different conditions, for example, given a fixed versus learned model of risk, various p-values, close-by-cliff world versus the far-away-cliff world as the environment.

The exploration methods are tested in the context of both Q-learning and Sarsa. The comparison of the performance of the exploration method

under Q-learning and Sarsa reflects the well-established fact that Q-learning generally yields steeper curves and higher end scores compared to Sarsa. Aside from this distinction, the results for Q-learning and Sarsa are qualitatively similar. Sarsa, which updates the action values based on the actions that the agent actually takes, is more realistic than Q-learning, and therefore, will be the focus in this analysis of results. All equivalent results for Q-learning can be found in the Appendix A and B.

5.2.1 Varying p-values

Figure 5–2 highlights the online learning behavior of the algorithm using different values of p and a fixed, a priori model.

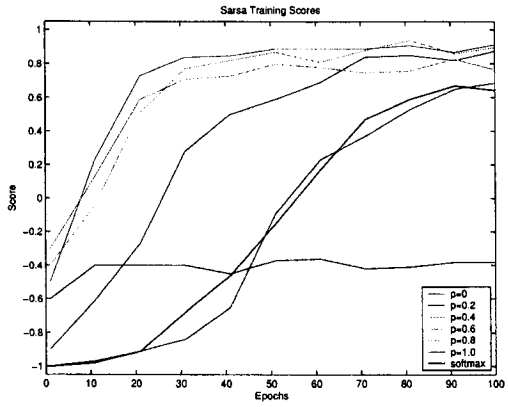
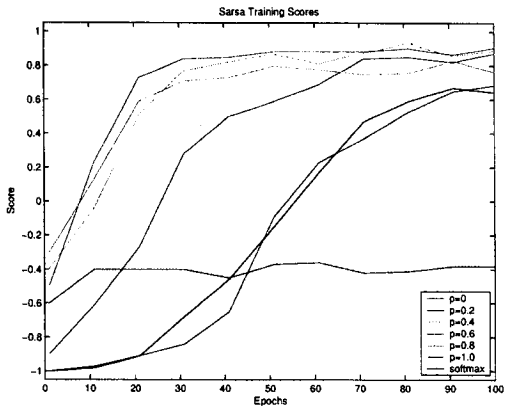


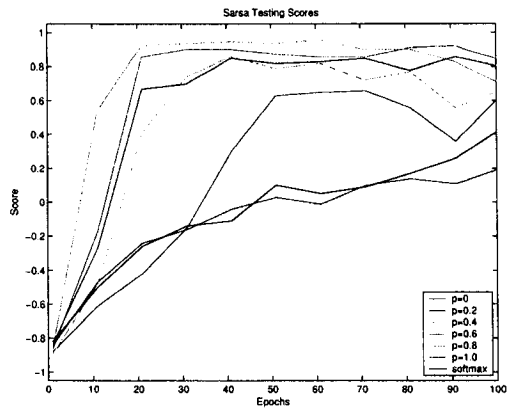
Figure 5–2: Close-by-cliff world: training score of Sarsa using fixed model

At $p = 1$, the agent only considers immediate risk in selecting an action. The training score shows a flat landscape, indicating that the agent wanders indefinitely (averaging 40000 time steps each episode, as shown in Figure 5–3(e)) until some stochasticity in the environment makes it fall off the cliff. Figure 5–3 shows that at $p = 1$, the agent's life terminates by cliff fall approximately 70% of the time during training (Figure 5–3(c)). The testing score, however, indicates that the agent learned a relatively good policy for

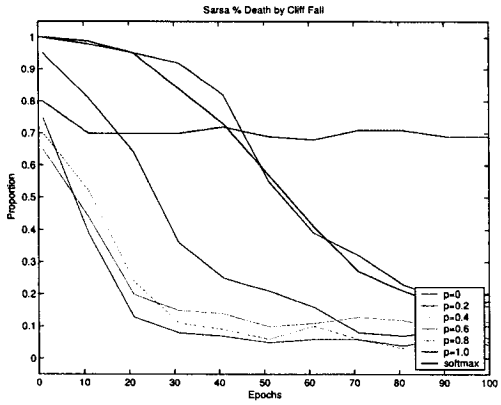
reaching the goal even for $p = 1$. This may be due to the significantly higher amount of sampling due to the long life span (approximately 35000 steps).



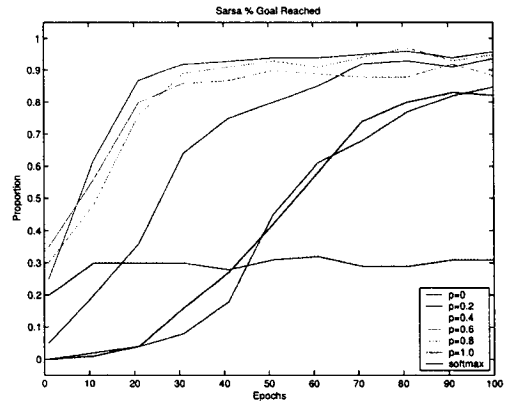
(a) Training Score



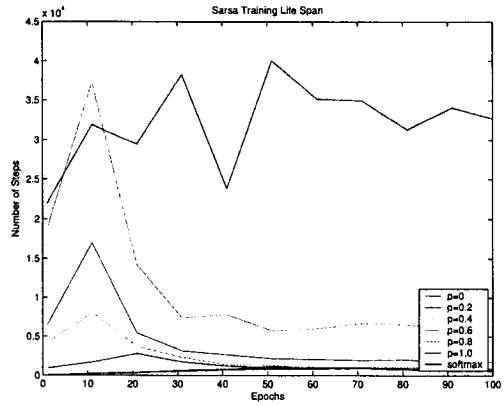
(b) Testing Score



(c) % Termination by Cliff Fall



(d) % Termination by Reaching Goal



(e) Life Span

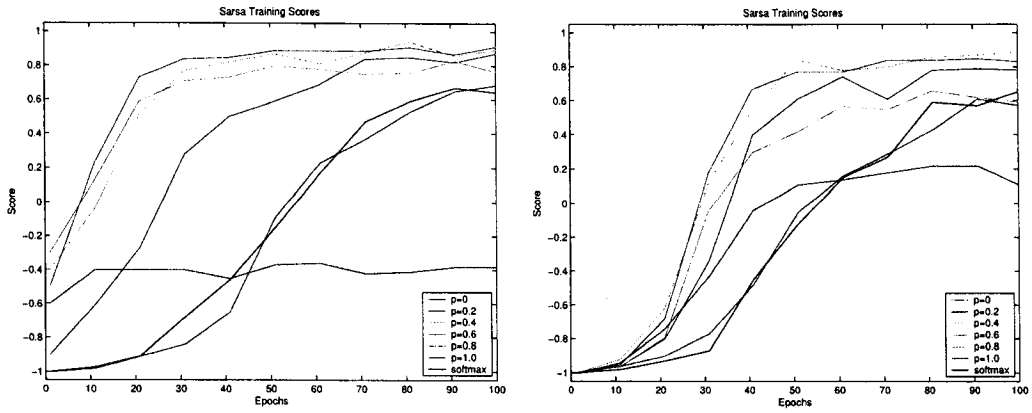
Figure 5–3: Close-by-cliff world: performance of Sarsa using fixed model

On the other end of spectrum, at $p = 0$, the agent only considers expected utilities in selecting an action. This is equivalent to the standard Boltzman exploration method. As clearly illustrated, the results for softmax and risk-directed exploration with $p = 0$ are qualitatively similar. The mid-range p-values outperform p-values in the extremes. In particular, under intermediate p-values, the training scores of risk-directed exploration are higher from the very beginning. This suggests that if the model of risk is perfectly known, the information contained in the model is capable of directing the agent towards a safe (low risk) path around the cliff, thereby reaching the goal more often. More complex risk averse behavior can be observed (Law *et al.*, 2005) when the agent is placed in a complex cliff world with multiple cliffs and goals.

5.2.2 Fixed versus Learned Model

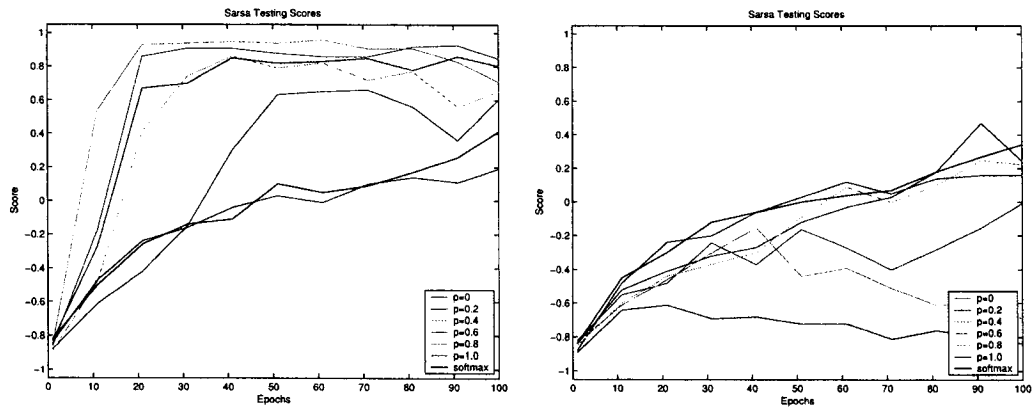
In order for the algorithm to be useful, it must be capable of learning the risk model online. Since an incorrect model is learned at the beginning and becomes more accurate only slowly through experience, the performance of the algorithm is generally lower than when given the fixed model. In fact, it takes twice as many trials for the algorithm to reach the same training score (Figure 5–4) using the learned model than using the fixed model. In addition, the testing score (Figure 5–5) for risk-directed exploration using the learned model is significantly worse than when the fixed model is used.

The model error (Figure 5–6) is computed over the course of the learning phase. It is observed that the model error is highest for $p = 1.0$ and $p = 0.0$, and lowest for the intermediate p-values. The model error for $p = 1.0$ is high possibly because the action values are not taken into account during the exploration. As a result, the agent consistently revisits the same states and updates the statistics for those states accurately, while completely



(a) Training Score for Fixed Model **(b)** Training Score for Learned Model

Figure 5–4: Close-by-cliff world: training scores of Sarsa using fixed versus learned model



(a) Testing Score for Fixed Model **(b)** Testing Score for Learned Model

Figure 5–5: Close-by-cliff world: testing scores of Sarsa using fixed versus learned model

ignoring other states. For $p = 0.0$, the reason for the higher model error is possibly the fact that the agent has a much shorter life span, thus less sampling, due to the comparatively higher number of cliff falls.

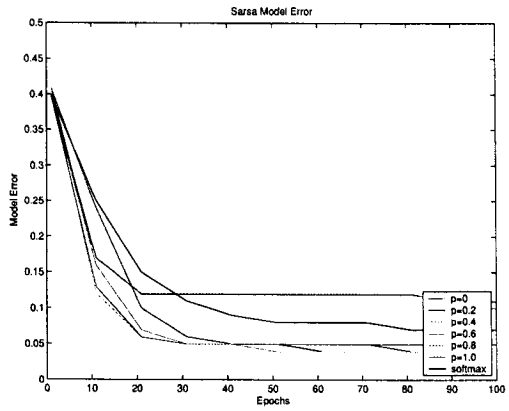


Figure 5–6: Close-by-cliff world: model error during training

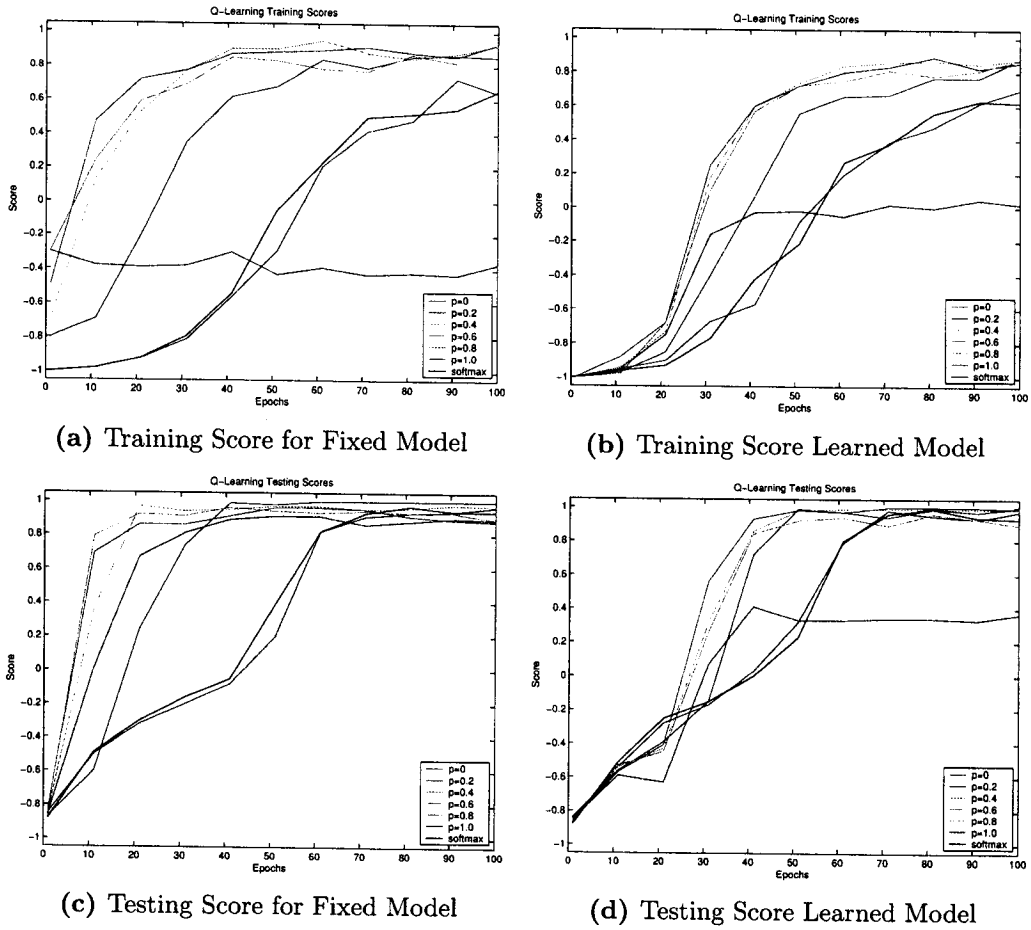
One final note is that the deterioration of performance for using a learned model is not nearly as marked for the same experiments done using Q-learning. In fact, both the training and testing score for Q-learning (Figure 5–7) and risk-directed exploration using the learned model remain superior to that of using Boltzmann exploration based only on action values.

5.2.3 Structure of Environment

Some qualitative differences in performance can be observed if we consider the far-away cliff world instead. Whereas in the close-by-cliff world, the policy learned by our algorithm can be worse than Boltzmann (Figure 5–5), in the far-away-cliff world, the policy is consistently better than Boltzmann (Figure 5–8) using both fixed and learned models.

In general, the training and testing scores for the far-away-cliff world are much higher than for the close-by-cliff world (Figure 5–9).

These results can be explained by the fact that in the close-by-cliff world, the agent has a higher probability of falling off the cliff due to environmental stochasticity. In contrast, in the far-away-cliff world, the agent is able to avoid the cliff almost entirely by going straight to the goal.



(a) Training Score for Fixed Model

(b) Training Score Learned Model

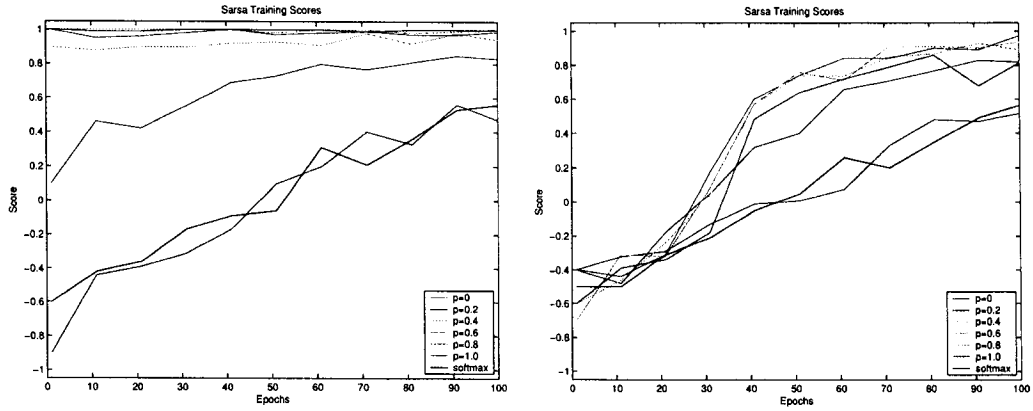
(c) Testing Score for Fixed Model

(d) Testing Score Learned Model

Figure 5–7: Close-by-cliff world: performance of Q-Learning using fixed versus learned model

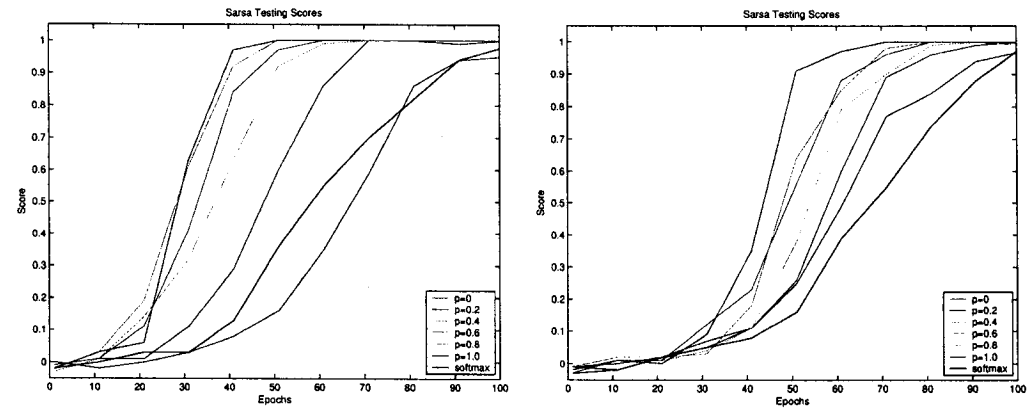
5.2.4 Comparison of Directed Exploration Methods

The following set of experimental results (Figure 5–10, 5–11, 5–12, 5–13) compare the performance of the risk-directed exploration method and that of the recency-based, counter-based, and standard Boltzmann exploration using just action values in different environments and using a fixed versus learned model. In general, recency-based and counter-based exploration yield higher training scores, but lower testing scores, than Botzmann exploration based on action values. The training and testing score of risk-directed exploration using $p = 0.6$ is consistently higher than recency-based, counter-based and standard Boltzmann, except for the case



(a) Training Score for Fixed Model

(b) Training Score for Learned Model

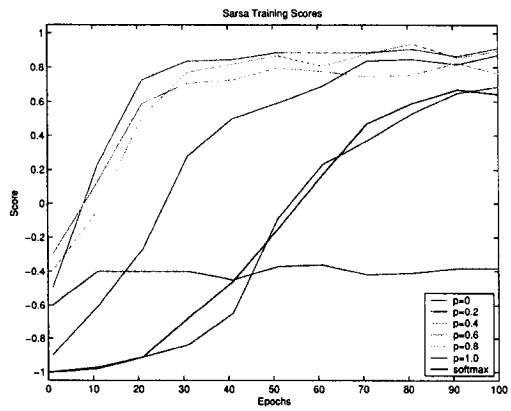


(c) Testing Score for Fixed Model

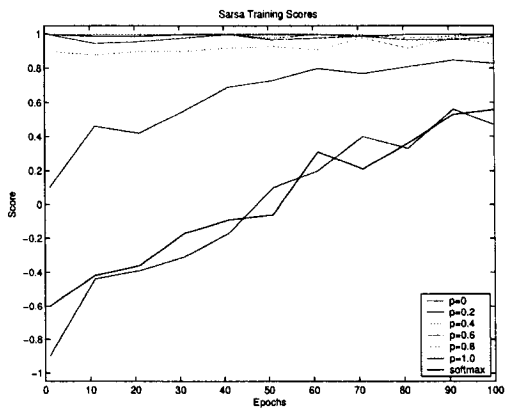
(d) Testing Score for Learned Model

Figure 5–8: Far-away-cliff world: performance of Sarsa using fixed versus learned model

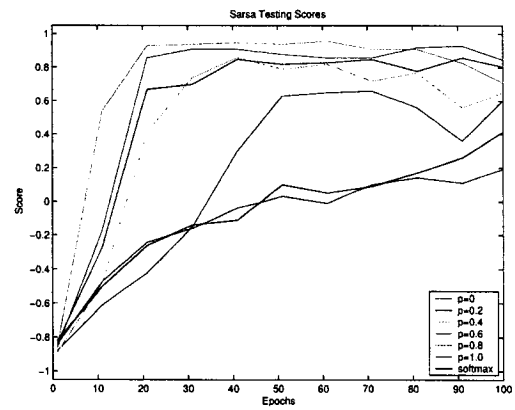
where the learned model is used in the close-by-cliff world. This improved learning performance can be explained by the fact that the percentage of cliff falls is significally lower during the learning phase and that the amount of sampling is increased significantly due to the longer life span secured by being risk-averse.



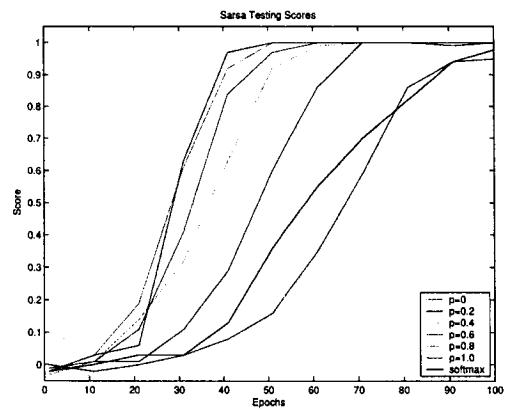
(a) Training score for close-by-cliff World



(b) Training score for far-away-cliff World

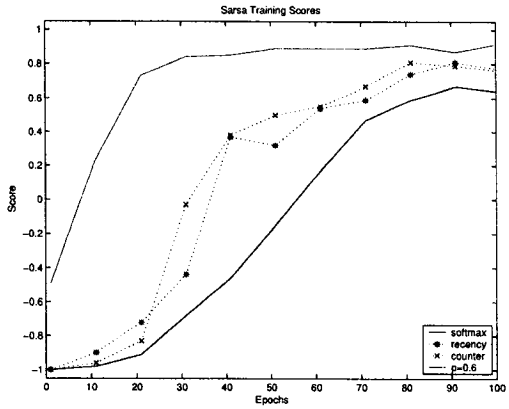


(c) Testing score for close-by-cliff World

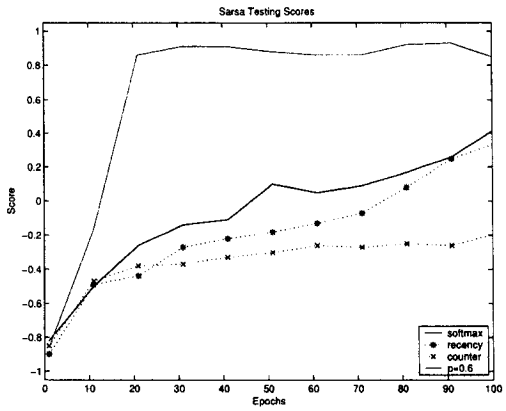


(d) Testing score for far-away-cliff World

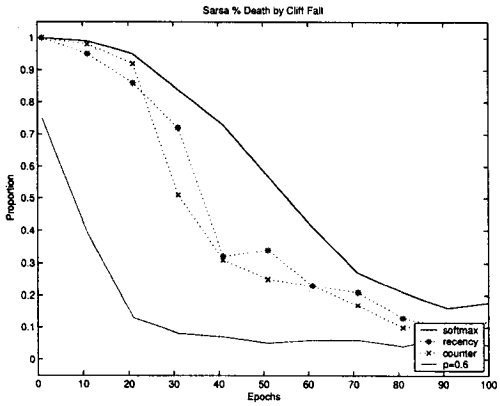
Figure 5–9: Close-by-cliff world versus far-away-cliff world: performance of sarsa using fixed model



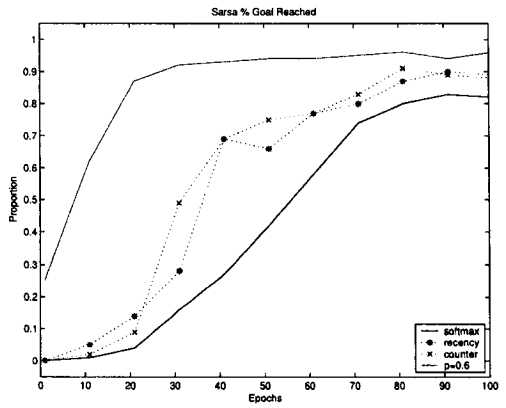
(a) Training Score



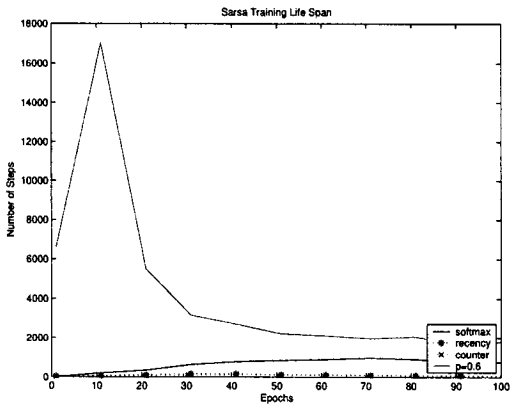
(b) Testing Score



(c) % Termination by Cliff Fall

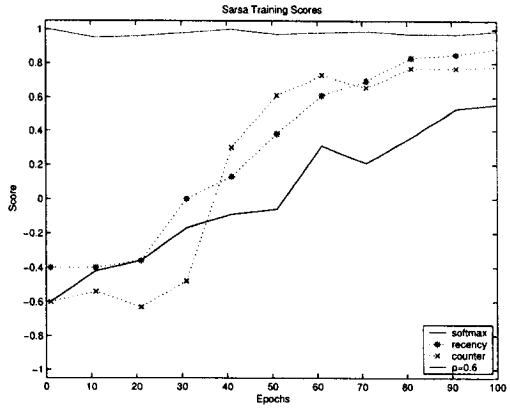


(d) % Termination by Reaching Goal

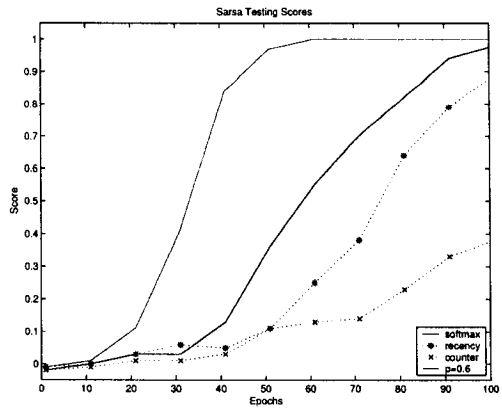


(e) Life Span

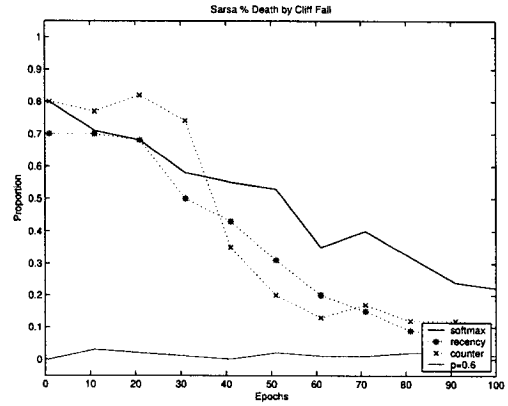
Figure 5–10: Comparison of the performance of Boltzmann, recency-based, counter-based, and risk-based exploration (p=0.6) method in close-by-cliff world using Sarsa and fixed model



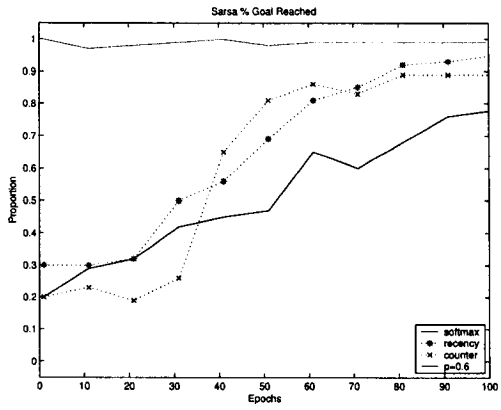
(a) Training Score



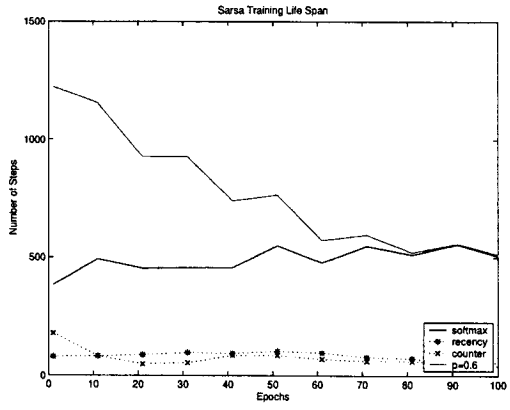
(b) Testing Score



(c) % Termination by Cliff Fall

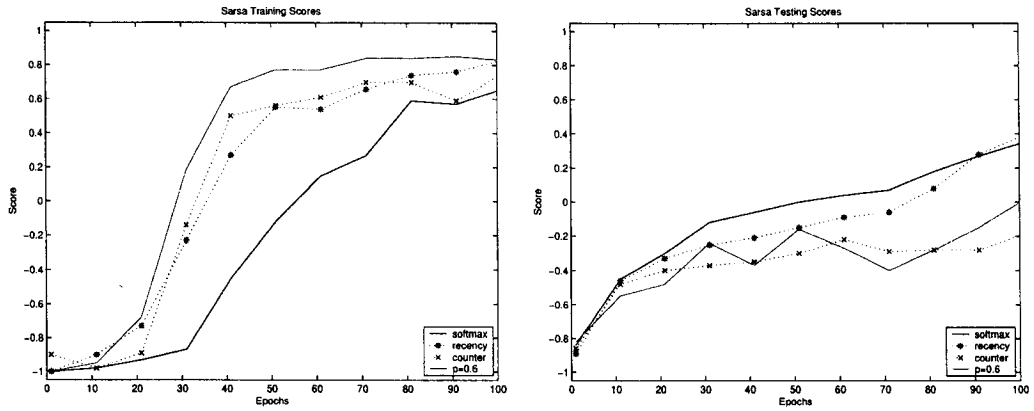


(d) % Termination by Reaching Goal



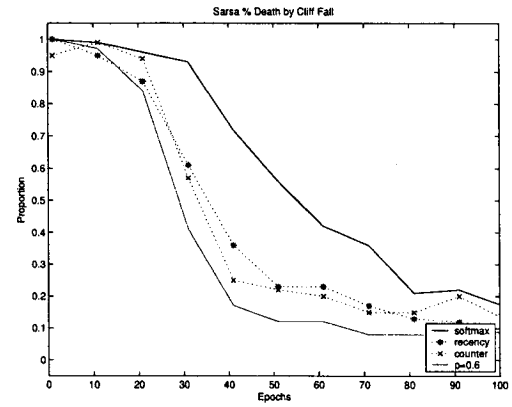
(e) Life Span

Figure 5–11: Comparison of the performance of Boltzmann, recency-based, counter-based, and risk-based exploration (p=0.6) method in far-away-cliff using Sarsa and fixed model

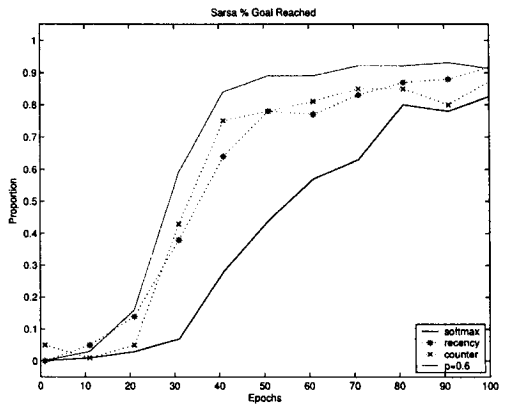


(a) Training Score

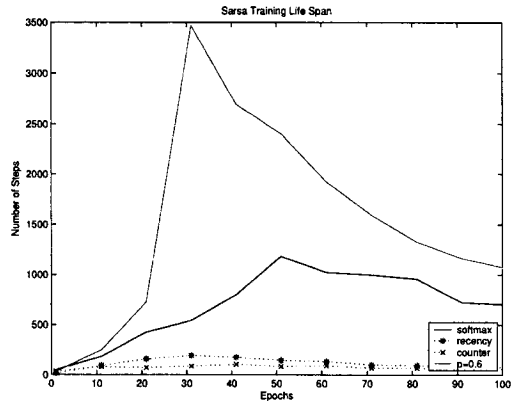
(b) Testing Score



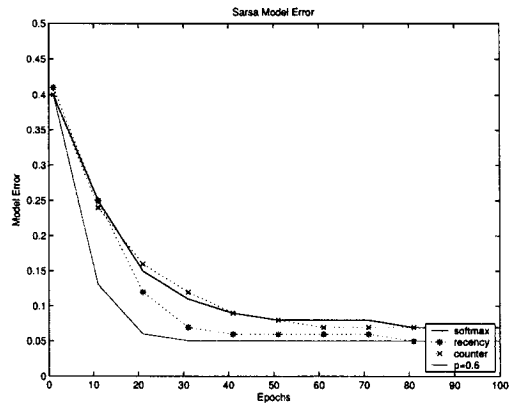
(c) % Termination by Cliff Fall



(d) % Termination by Reaching Goal

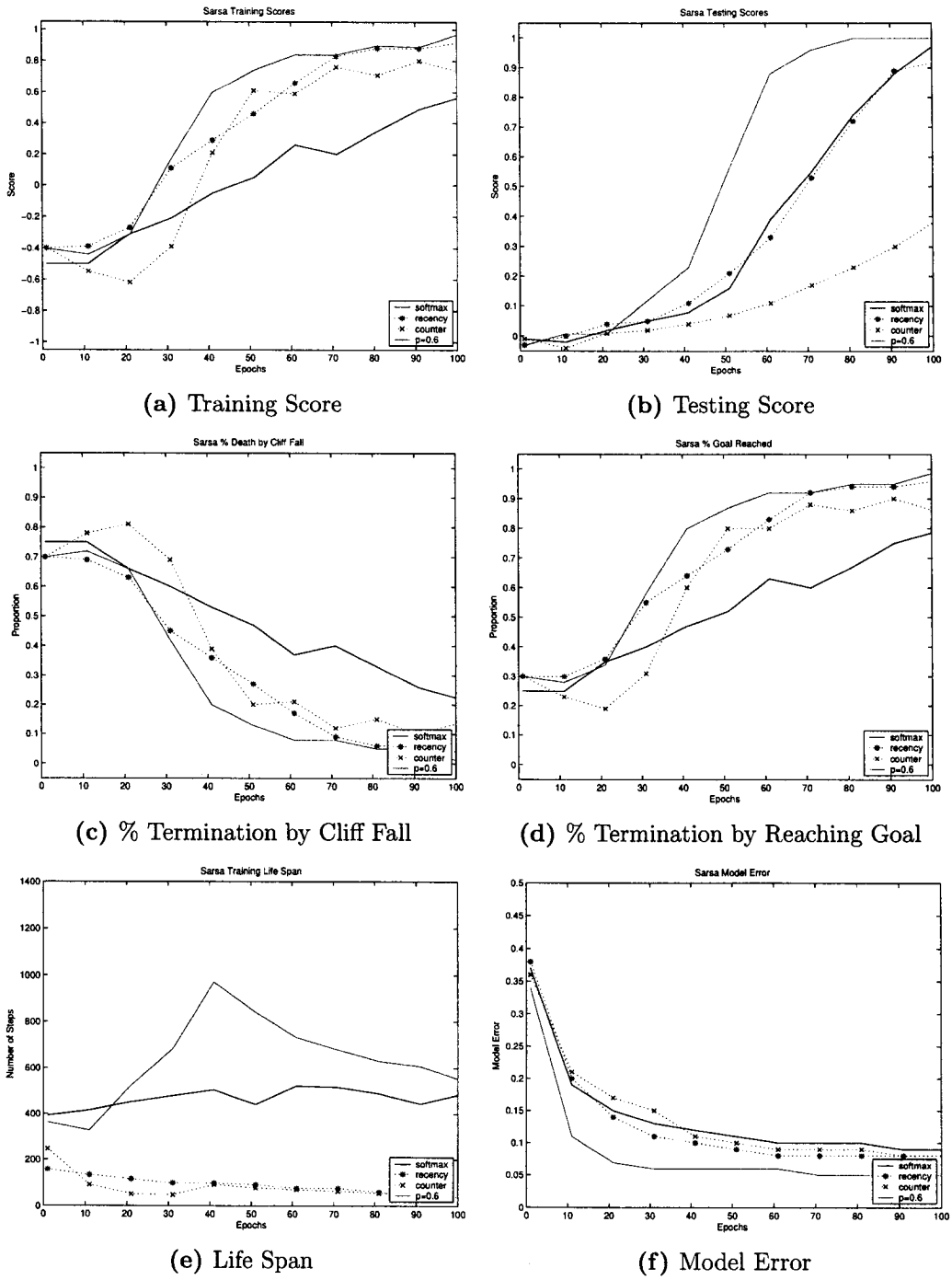


(e) Life Span



(f) Model Error

Figure 5–12: Comparison of the performance of Boltzmann, recency-based, counter-based, and risk-based exploration (p=0.6) method in close-by-cliff using Sarsa and learned model



(a) Training Score

(b) Testing Score

(c) % Termination by Cliff Fall

(d) % Termination by Reaching Goal

(e) Life Span

(f) Model Error

Figure 5–13: Comparison of the performance of Boltzmann, recency-based, counter-based, and risk-based exploration (p=0.6) method in far-away-cliff using Sarsa and learned model

CHAPTER 6
Conclusions

The dangers of life are infinite, and among them is safety.
- Goethe

The risk-directed exploration method presented here offers a simple and intuitive solution for preserving survival during learning by risk avoidance. The mechanism of risk avoidance is achieved by learning the risk values of actions during learning, based on which the probability of an action being selected is adjusted. Our experimental results show that the training and testing score are in general higher than other directed exploration methods, especially during the early stages of learning.

One criticism of this method may be that by visiting only states that are less risky, the agent does not sample widely enough to have an accurate picture of the environment. As a result, the policy that is learned may be suboptimal. While this is a perfectly valid argument, our standpoint is that if the preservation of survival is one of the criteria of an efficient exploration method, then we can sacrifice some optimality in order to keep the agent safe. For values of the parameter $p < 1$, optimality can still be achieved in the limit. This feature is not present in other algorithms which change the action values being learned.

Similarly, the claim that risk aversion is useful for survival is likely to provoke disagreement. One may argue that risk aversion is useful in certain situations, but it can produce pathological behavior in others. Imagine a cliff world environment where the cliff divides the space between the agent and the cheese. The risk-directed exploration method will select actions

such that the agent remains in the *safer* region of the environment, never approaching the cheese, and hence, eventually running out of resources. Being risk-averse, in this case, does not guarantee survival.

Reflection on the limitation of risk aversion suggests that it may be beneficial for the agent to be risk-averse at certain times, but risk-seeking at other times depending on the current context. In fact, risk sensitivity in decision making has been widely observed in the study of animal foraging behaviour. In one experiment, *Junco phaeonotus*, or yellow-eyed junco birds, were presented with a choice between a feeding station that provides a constant supply of three seeds and a second feeding station that provides either no seeds or six seeds with equal probability. It is found that the birds' preferences for the two foraging options depended on the temperature. At normal temperature (19°C), the birds are on a positive energy budget, i.e. the average reward of three seeds is sufficient to maintain the energy level above a critical threshold. It is observed that the birds prefer the constant foraging option that provides three seeds, i.e. they are risk-averse. At low temperature (1°C), where the average reward of three seeds can no longer compensate for the energy expenditure, a reversal in the preference is observed. The birds were risk-seeking, preferring the variable foraging option that has some probability of providing enough seeds to bring the energy level above the critical threshold (Caraco *et al.*, 1990). This switch between risk-seeking and risk-averse behavior is also observed when the source of hazard is not resource depletion, but predation (Milinski and Heller, 1978).

These observations of animal foraging behavior have interesting implication for decision making in uncertain environments. First, this evidence supports the fact that a measure of risk, instead of expected utility, can be potentially useful for the valuation of a prospect. Second, the ability

to adjust risk attitude depending on context seems to have a clear advantage in ensuring survival, and is empirically shown to exist even in human decision making (March and Shapira, 1992). In addition, risk-sensitivity may be useful also for modelling a wide range of emotions, behavior and personality in the agent.

The risk-directed exploration presented in this thesis can be easily extended to provide a framework in which the risk attitude is dynamically altered during the learning phase based on the current context. This can be done by adjusting the p-value subject to some predetermined schedule of decay, or according to some other constraints. In this thesis, we focus on understanding the behavior and performance of the risk-directed exploration method for a fixed level of p-value. Hence, the appropriate mechanisms for dynamically controlling the risk attitude remains an open research question.

There are many other possible extensions to this work. In Chapter 4, we note that using the risk adjusted utility in the Boltzmann function actually boosts the probability of selecting an action with low action values, a property that is counter-intuitive to what risk aversion means. What is desired, instead, is a function which depresses the probability of selecting *good* actions according to their risk values, but also keeps the probability of selecting bad actions low. The function resembles the utility function for decision making posed by the Propsect Theory, which is concave for gain and convex for losses (as seen in Figure 6–1).

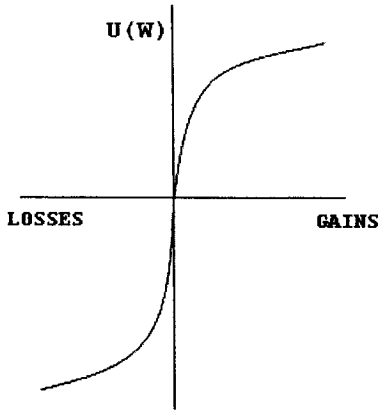


Figure 6–1: Prospect theory

The empirical evidence for Prospect Theory comes from the well-known Allais Paradox (Allais, 1953), which can be illustrated by the following example (Kahneman and Tversky, 1979).

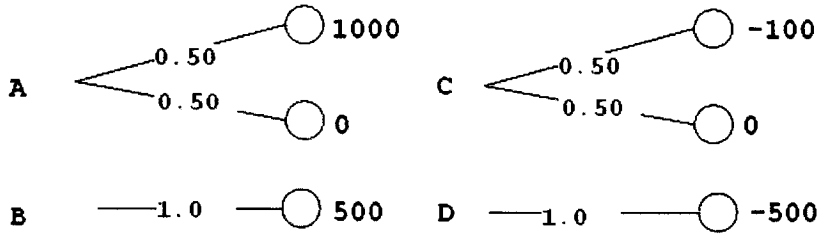


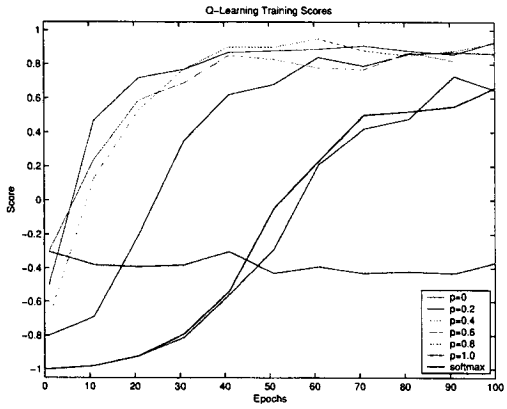
Figure 6–2: Allais paradox

Subjects were asked to do two independent choice experiments. In one case, they have to choose between receiving 1000 or nothing with equal probability of 0.5 (Choice A), or receiving 500 for certain (Choice B). In the second case, they have to choose between losing 1000 or nothing with equal probability of 0.5 (Choice C), or losing 500 for sure (Choice D). The majority of the subjects choose B in the first case and choose C in the second. This suggests that people are risk-averse when prospect are

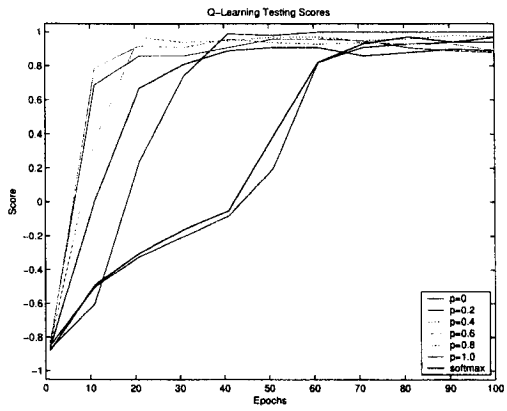
framed in terms of gain, and risk-seeking when prospects are framed in terms of losses. This is the reason why the corresponding Prospect Theory utility function is concave for gain and convex for losses. The possibility of modifying the Boltzmann action selection curve to induce risk-seeking behavior when negative utility are involved remains to be investigated.

Another interesting enhancement would be the use of a multi-step risk measure where the $step > 2$. An adjustable window of how far to look ahead when calculating risk can be analogous to paying attention to short term, medium term, or long term risk of an action. Second, the risk measure can be subject to TD learning so that a global, instead of local, measure of risk is derived. In order to subject the risk measure to dynamic programming, the risk measure must have certain desirable properties, for example, additivity. Hence, it would be useful to characterize exactly what those desirable properties are, and what other definitions of risk are suitable for application. Finally, the current approach directs exploration using a local measure of risk that is centered around the state-action pair. One possibility is to use of a hierarchical approach (Dayan and Hinton, 1993) to learn, at a much higher level, which *regions* of the state space are risky.

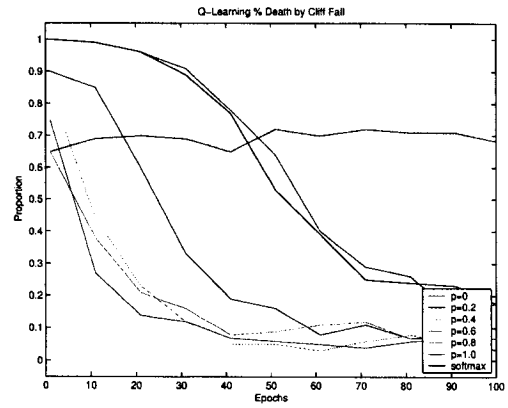
Appendix A: Performance of Risk-Directed Exploration for Q-Learning



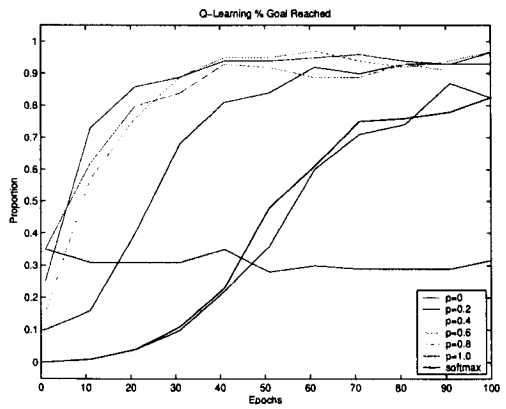
(a) Training Score



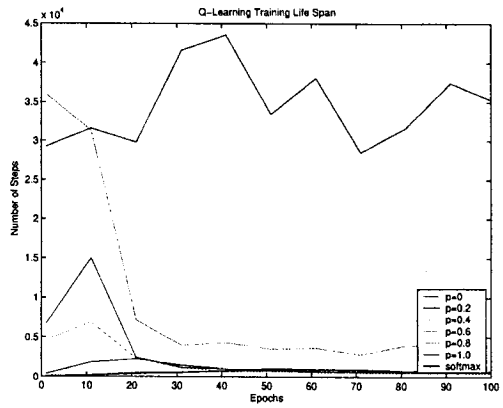
(b) Testing Score



(c) % Termination by Cliff Fall

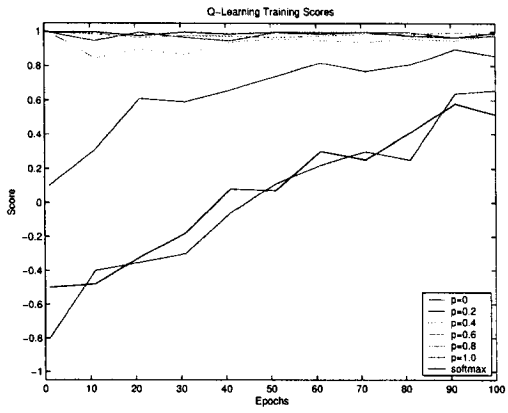


(d) % Termination by Reaching Goal

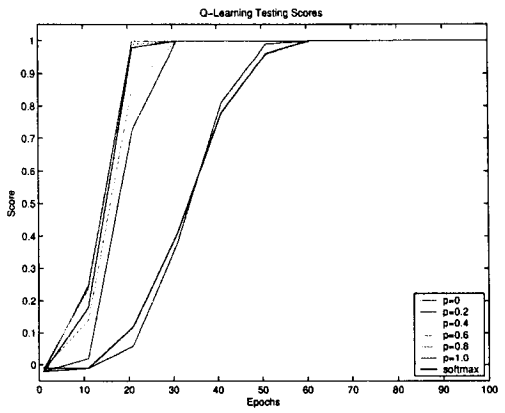


(e) Life Span

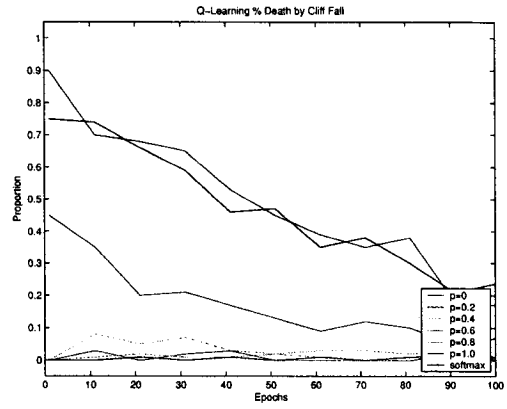
Figure 6–3: Close-by-cliff world: performance of Q-Learning using fixed model



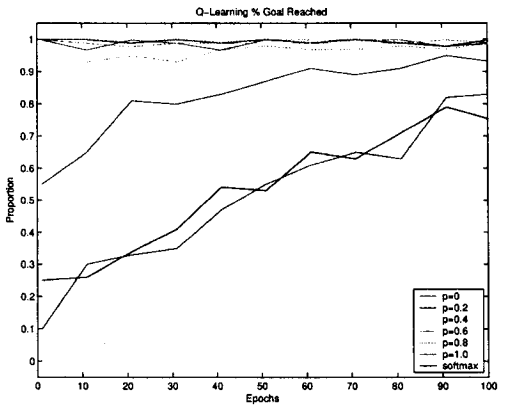
(a) Training Score



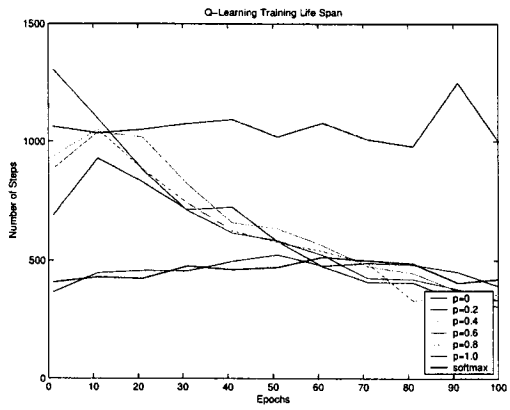
(b) Testing Score



(c) % Termination by Cliff Fall

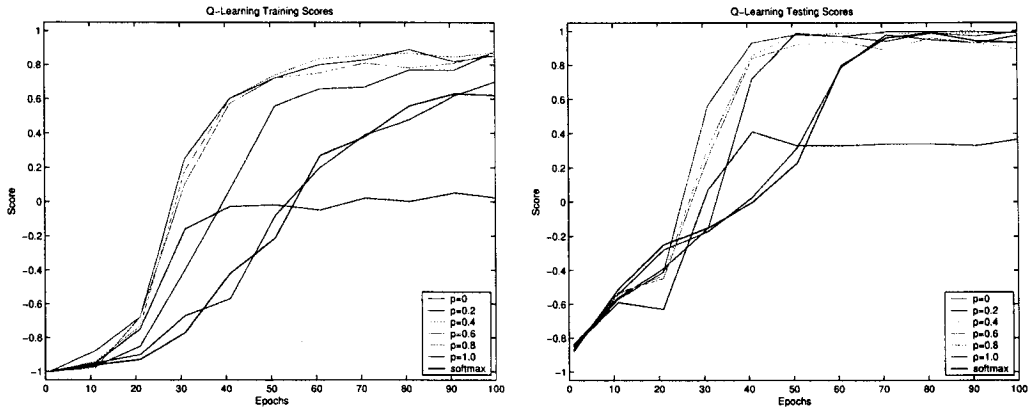


(d) % Termination by Reaching Goal



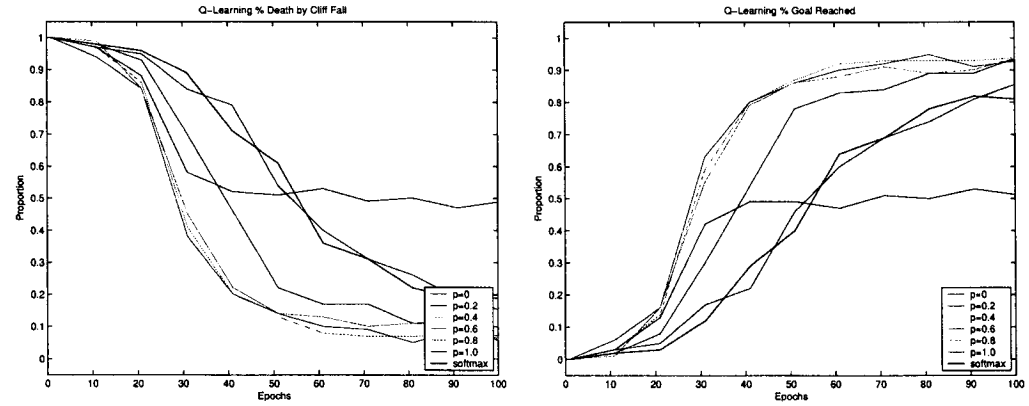
(e) Life Span

Figure 6–4: Far-away-cliff world: performance of Q-Learning using fixed model



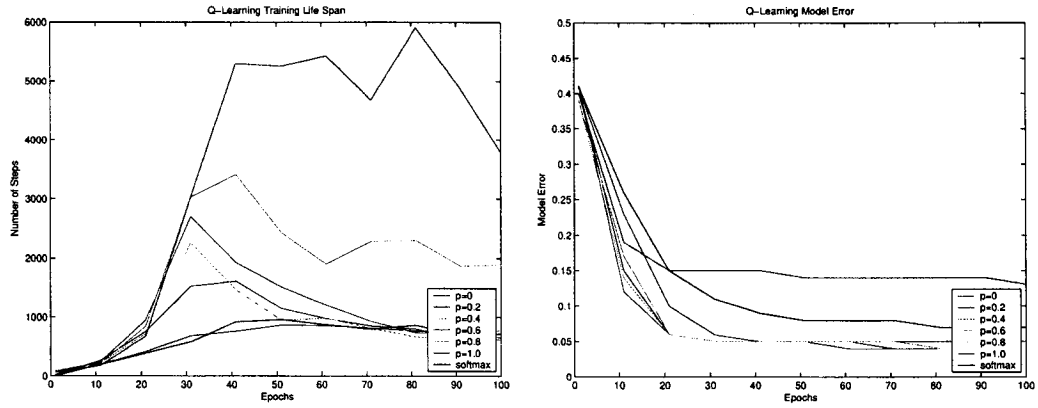
(a) Training Score

(b) Testing Score



(c) % Termination by Cliff Fall

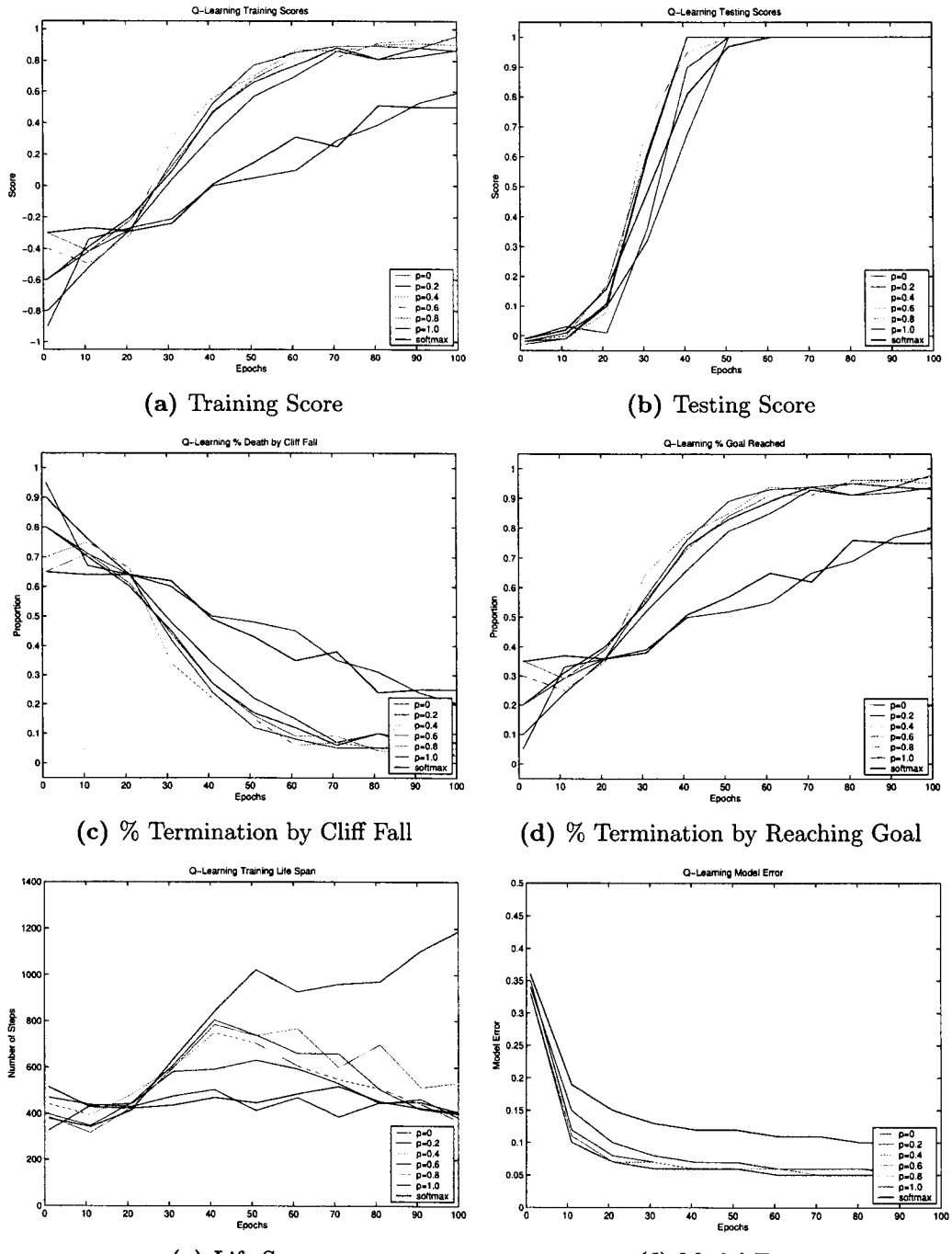
(d) % Termination by Reaching Goal



(e) Life Span

(f) Model Error

Figure 6–5: Close-by-cliff world: performance of Q-Learning using learned model



(a) Training Score

(b) Testing Score

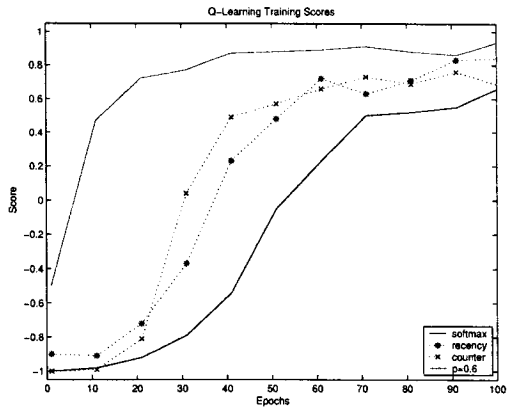
(c) % Termination by Cliff Fall

(d) % Termination by Reaching Goal

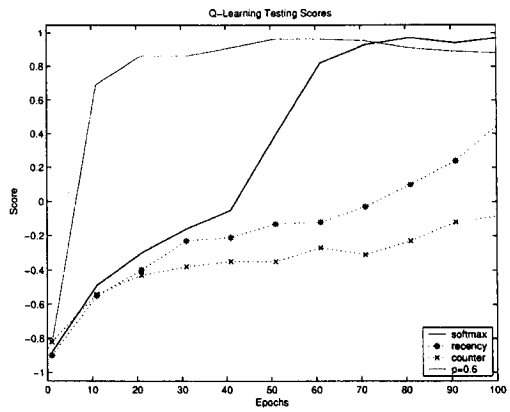
(e) Life Span

(f) Model Error

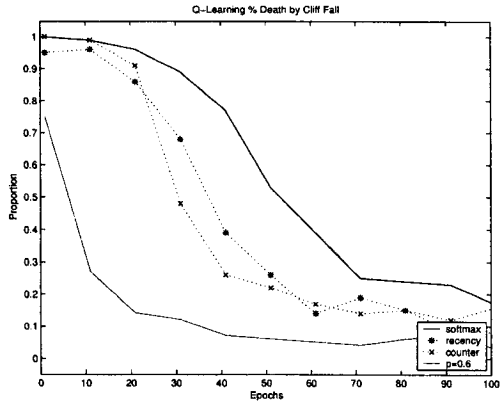
Figure 6–6: Far-away-cliff world: performance of Q-Learning using learned model



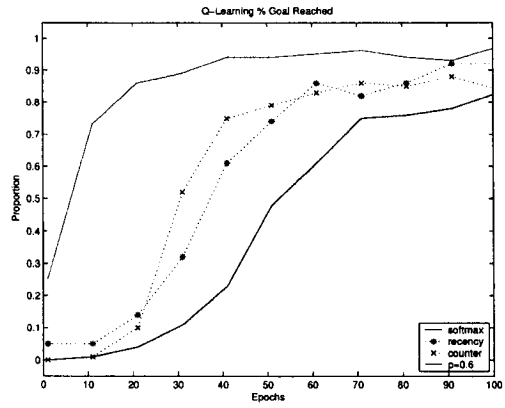
(a) Training Score



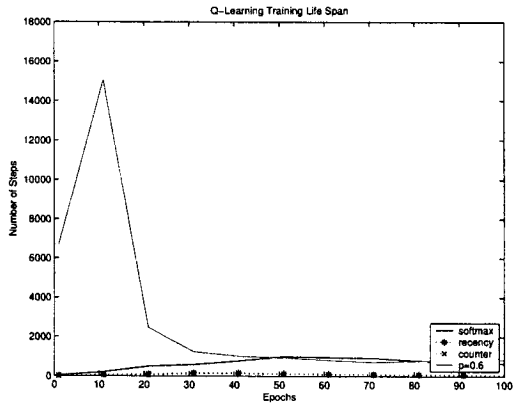
(b) Testing Score



(c) % Termination by Cliff Fall

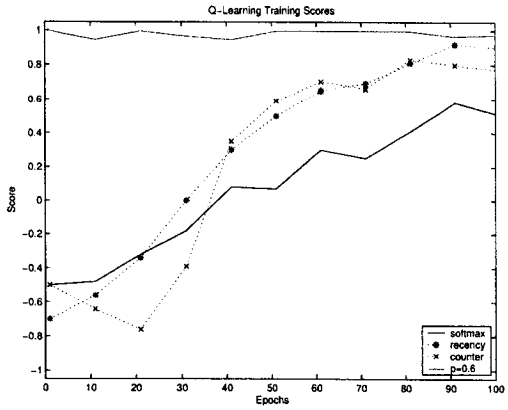


(d) % Termination by Reaching Goal

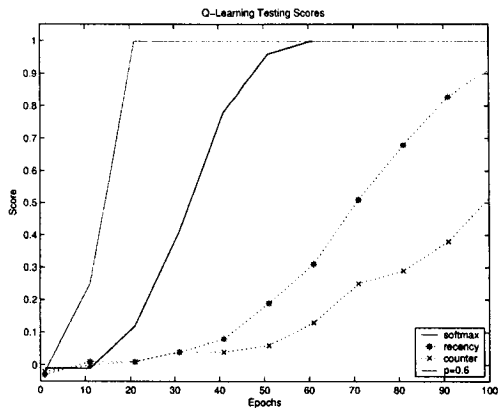


(e) Life Span

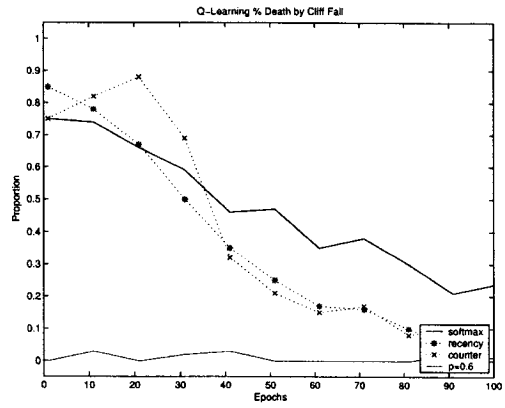
Figure 6–7: Comparison of the performance of Boltzmann, recency-based, counter-based, and risk-based exploration (p=0.6) method in close-by-cliff world using Q-Learning and fixed model



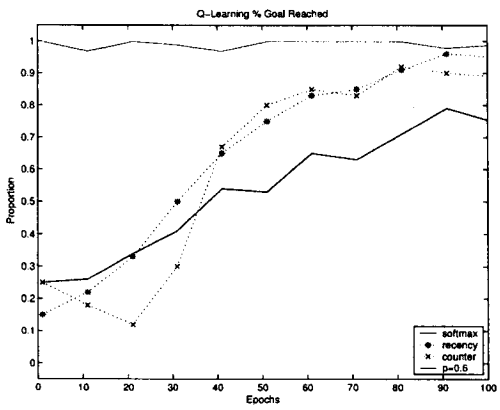
(a) Training Score



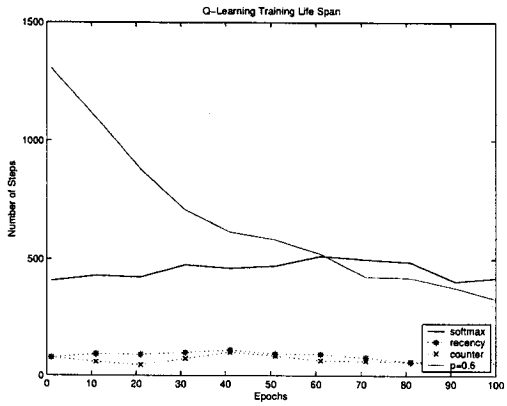
(b) Testing Score



(c) % Termination by Cliff Fall

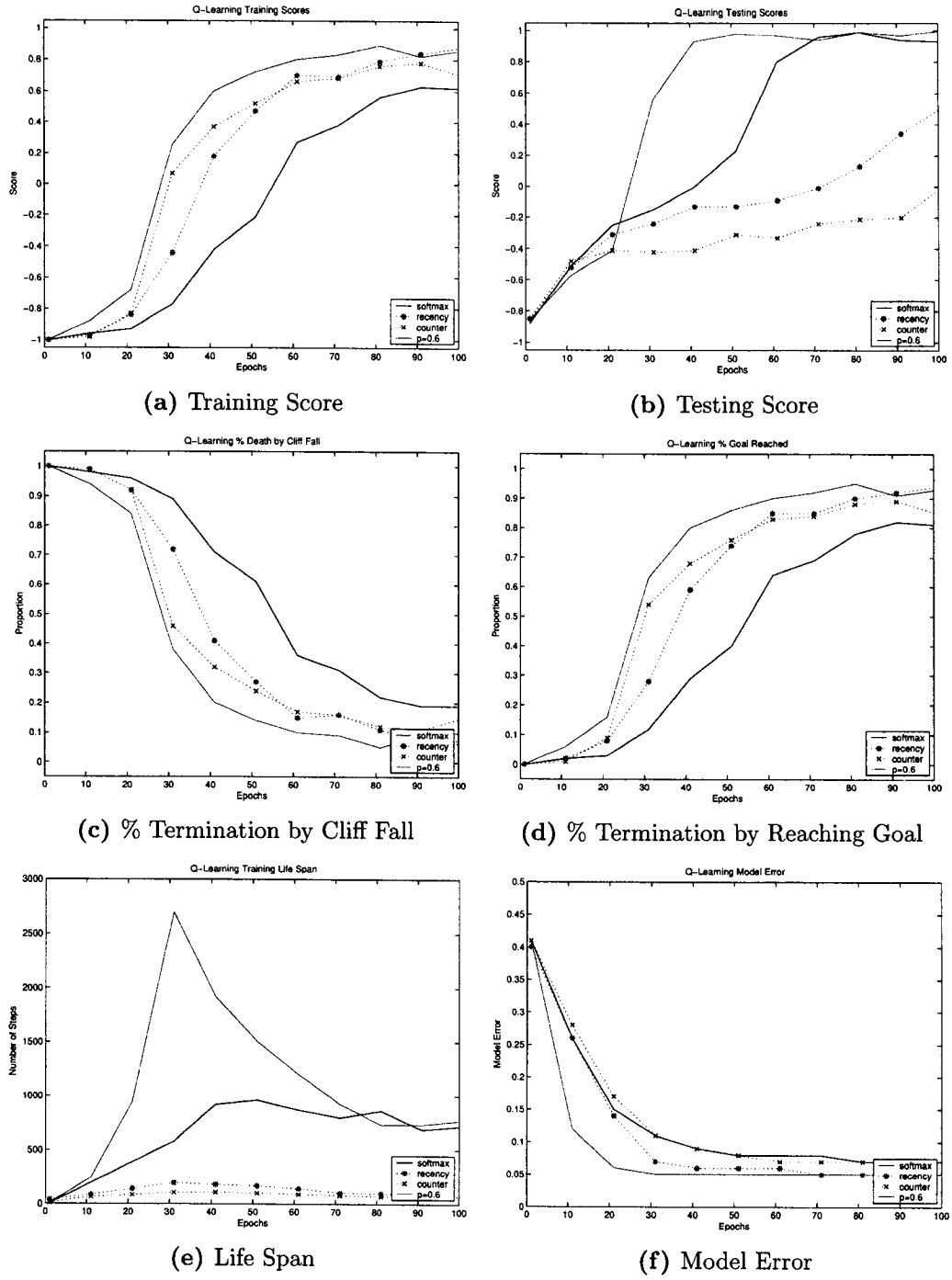


(d) % Termination by Reaching Goal



(e) Life Span

Figure 6–8: Comparison of the performance of Boltzmann, recency-based, counter-based, and risk-based exploration (p=0.6) method in far-away-cliff world using Q-Learning and fixed model



(a) Training Score

(b) Testing Score

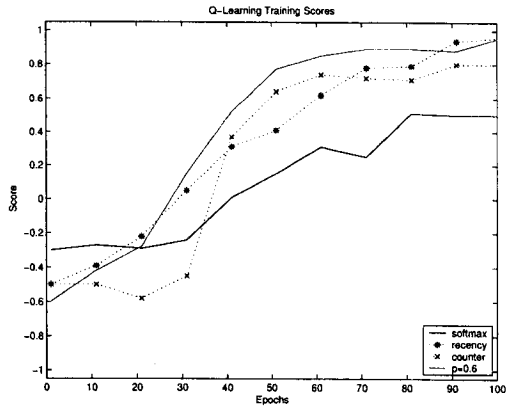
(c) % Termination by Cliff Fall

(d) % Termination by Reaching Goal

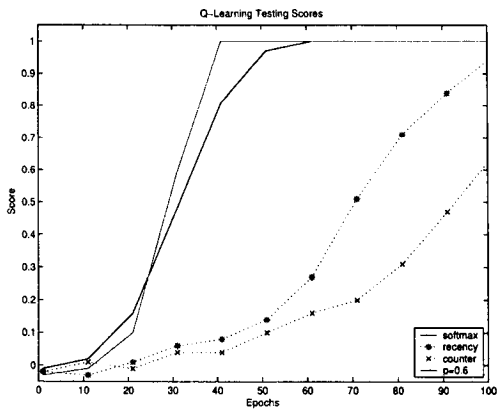
(e) Life Span

(f) Model Error

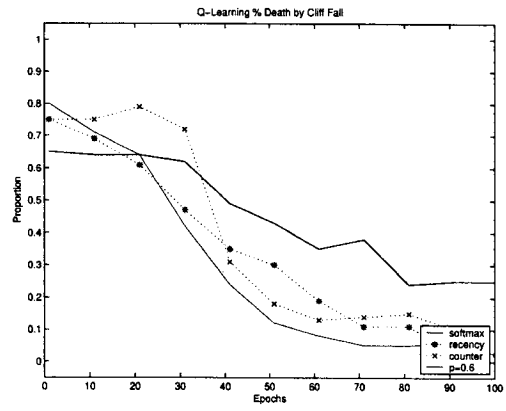
Figure 6–9: Comparison of the performance of Boltzmann, recency-based, counter-based, and risk-based exploration (p=0.6) method in close-by-cliff using Q-Learning and learned model



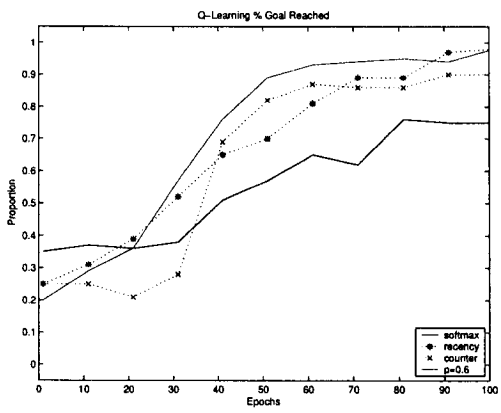
(a) Training Score



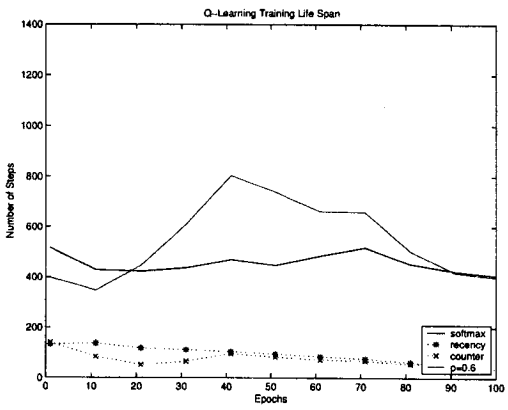
(b) Testing Score



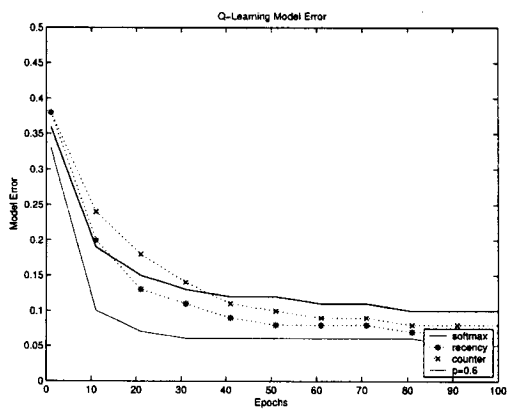
(c) % Termination by Cliff Fall



(d) % Termination by Reaching Goal



(e) Life Span



(f) Model Error

Figure 6–10: Comparison of the performance of Boltzmann, recency-based, counter-based, and risk-based exploration (p=0.6) method in close-by-cliff using Q-Learning and learned model

Bibliography

ALLAIS, M. (1953). Le comportement de l'homme, rationnel devant le risque, critique des postulats et axiomes de l'ecole americaine. *Econometrica*, 21:503–546.

BELLMAN, R. (1957). *Dynamic Programming*. Princeton University Press, Princeton, N.J.

BERRY, D. AND FRISTEDT, B. (1985). *Bandit Problems: Sequential Allocation of Experiments*. Chapman and Hall, London.

BRAFMAN, R. AND TENNENHOLTZ, M. (2002). R-max-a general polynomial time algorithm for near-optimal reinforcement learning. *Journal of Machine Learning Research*, 3:213–231.

BULITKO, V. (2004). Rl for life notes.

CARACO, T., BLANCKENHORN, W., GREGORY, G., NEWMAN, J., RECER, G., AND ZWICKER, S. (1990). Risk-sensitivity: ambient temperature affects foraging choice. *Animal Behavior*, 39:338–345.

CHRISTIANSEN, A., MASON, M., AND MITCHELL, T. (1991). Learning reliable manipulation strategies without initial physical models. *Robotics and Autonomous Systems*, 8:7–18.

DAYAN, P. AND HINTON, G. (1993). Feudal reinforcement learning. In LIPPMAN, D., MOODY, J., AND TOURETZKY, D., editors, *Advances in Neural Information Processing Systems 5*, pages 271–278. San Mateo, CA: Morgan Kaufmann.

DAYAN, P. AND SEJNOWSKI, T. (1996). Exploration bonuses and dual control. *Machine Learning*, 25:5–22.

DEARDEN, R., FRIEDMAN, N., AND ANDRE, D. (1999). Model-based bayesian exploration. In *Proceedings of the 15th UAI Conference*, pages 150–159.

FELDBAUM, A. (1960). Dual control theory i. *Automn Remote Control*, pages 874–880.

FILAR, J., KALLENBERG, L., AND LEE, H. (1989). Variance-penalized markov decision processes. *Mathematical Operations Research*, 14:147–161.

GASKETT, C. (2003). Reinforcement learning under circumstances beyond its control. In *Proceedings of the International Conference on Computational Intelligence for Modelling Control and Automation.*

GEIBEL, P. (2001). Reinforcement learning with bounded risk. In BRODLEY, C. E. AND DANYLUK, A., editors, *Proceedings of the Eighteenth International Conference (ICML01)*, pages 162–169, San Francisco,CA. Morgan Kaufmann Publishers.

GITTINS, J. C. (1998). *Multi-armed Bandit Allocation Indices.* Wiley-Interscience series in systems and optimization. Wiley.

HEGER, M. (1994). Consideration of risk in reinforcement learning. In *Proceedings of the eleventh International Conference on Machine Learning*, pages 105–111.

HERNANDEZ-HERNANDEZ, D. AND MARCUS, S. (1966). Risk-sensitive control of markov processes in countable state space. *Systems and Control Letters*, 29:147–155.

HOWARD, R. AND MATHESON, J. (1972). Risk-sensitive markov decision processes. *Management Sciences*, 8:356–369.

HUANG, Y. AND KALLENBERG, L. (1994). On finding optimal policies for markov decision chains: a unifying framework for mean-variance

tradeoffs. *Mathematical Operations Research*, 19:434–448.

KAELBLING, L. (1993). *Learning in Embedded Systems.* The MIT Press.

KAHNEMAN, D. AND TVERSKY, A. (1979). Prospect theory: An analysis of decision under risk. *Econometrica*, 47(2):263–292.

KEARNS, M. J. AND SINGH, S. P. (2002). Near-optimal reinforcement learning in polynomial time. *Machine Learning*, 49:209–232.

KOENIG, S. AND SIMMONS, K. (1994). Risk sensitive planning with probabilistic graphs. In *Proceedings of the Fourth International Conference on Principles of Knowledge Representation and Reasoning (KR-94), Bonn, Germany*, pages 363–373.

KRUUSMAA, M. (1999). Decision-making for autonomous robots in hazardous environments. In *Proceedings of the IASTED International Conference of Robotics and Applications*, pages 244–249.

LAW, E., COGGAN, M., PRECUP, D., AND RATITCH, B. (2005). Risk-directed exploration in reinforcement learning. In *IJCAI Workshop on Planning and Learning in A Priori Unknown or Dynamic Domains.*

MARCH, J. AND SHAPIRA, Z. (1992). Variable risk preferences and the focus of attention. *Psychological Review*, 99(1):172–183.

MARKOWITZ, H. (1952). Portfolio selection. *Journal of Finance*, 7:77–91.

MEULEAU, N. AND BOURGINE, P. (1999). Exploration of multi-state environment: Local measure and back-propagation of uncertainty. *Machine Learning*, 35:117–154.

MILAN, J. (1996). Rapid, safe and incremental learning of navigation strategies. In *Proceedings of the IEEE Transactions on Systems, Man, and Cybernetics*, volume 26(3), pages 408–420.

MILINSKI, M. AND HELLER, R. (1978). Influence of a predator on the optimal foraging behavior of stickleback. *Nature*, 275:642–644.

MOORE, A. (1990). *Efficient Memory-based Learning for Robot Control.* Ph.D. thesis, Trinity Hall, University of Cambridge, England.

NARENDRA, K. AND THATHACHAR, M. (1989). *Learning Automata: An Introduction.* Prentice-Hall, Englewood Cliffs, NJ.

NEUNEIER, R. AND MIHATSCH, O. (2002). Risk-sensitive reinforcement learning. *Machine Learning*, 49:267–290.

PRATT, J. (1964). Risk aversion in the small and in the large. *Econometrica*, 32(1-2):122–136.

PUTERMAN, M. (1994). *Markov Decision Processes - Discrete Stochastic Dynamic Programming.* John Wiley and Sons, Inc.

RATITCH, B. AND PRECUP, D. (2003). Using mdp characteristics to guide exploration in reinforcement learning. In *Proceedings of the ECML-2003.*

SATO, M., ABE, K., AND TAKEDA, H. (1988). Learning control of finite markov chains with an explicit trade-off between estimation and control. In *Proceedings of the IEEE Transactions on Systems, Man, and Cybernetics*, volume 18(5), pages 677–684.

SATO, M. AND KOBAYASHI, S. (2000). Variance-penalized reinforcement learning for risk-averse asset allocation. In LEUNG, K., CHAN, L., AND MENG, H., editors, *IDEAL 2000*, pages 244–249.

SCHMIDHUBER, J. (1991). Adaptive confidence and adaptive curiosity. Report, FKI-149-91, Technische Universitat Munchen.

SINGH, S., BARTO, A., GRUPEN, R., AND CONNOLLY, C. (1994). Robust reinforcement learning in motion planning. In COWAN, J., TESAURO, G., AND ALSPECTOR, J., editors, *Advances in Neural Information Processing Systems 6*, pages 655–662. San Mateo, CA: Morgan Kaufmann.

SOBEL, M. (1982). The variance of a discounted markov decision process. *Journal of Applied Probability*, 19:794–802.

SUTTON, R. (1990). Integrated architectures for learning, planning and reacting based on dynamic programming. In *Proceedings of the Seventh International Conference on Machine Learning*, pages 216–224.

SUTTON, R. AND BARTO, A. (1998). *Reinforcement Learning: An Introduction.* The MIT Press.

SZEGO, G. (2004). Measures of risk. *European Journal of Operation Research*, pages 1–15.

TAHA, H. (1992). *Operations Research.* Prentice-Hall, Inc, Upper Saddle River, New Jersey.

THRUN, S. (1992a). Efficient exploration in reinforcement learning. Report, January, CMU-CS-92-102, Carnegie Mellon University, Pittsburgh, PA 15213.

THRUN, S. (1992b). The role of exploration in learning control. In WHITE, D. AND SOFGE, D., editors, *Handbook of Intelligent Control: Neural, Fuzzy, and Adaptive Approaches*, pages 527–559. Van Nostrand Reinhold.

THRUN, S. AND MOLLER, K. (1992). Active exploration in dynamic environment. In MOODY, J. E. A., editor, *Advances in Neural Information Processing 4*. San-Mateo, CA: Morgan Kaufmann.

WATKINS, C. AND DAYAN, P. (1992). Q-learning. *Machine Learning*, 8(3):279–292.

WHITE, D. (1992). Computational approaches to variance penalized markov decision processes. *OR Spektrum*, 14:79–83.

WHITE, D. (1994). A mathematical programming approach to a problem

in variance penalized markov decision processes. *OR Spektrum*, 15:225–230.

WILSON, S. (1996). Explore/exploit strategies in autonomy. In MEYER, J. AND WILSON, S., editors, *Proceedings of the Fourth International Conference on Simulation of Adaptive Behavior: From Animals to Animats 4*, pages 325–332. The MIT Press/Bradford Books.

WYATT, J. (2001). Exploration control in reinforcement learning using optimistic model selection. In *Proceedings of the Eighteenth International Conference on Machine Learning*, pages 593–600.

YANG, J. AND QIU, W. (2005). A measure of risk and a decision-making model based on expected utility and entropy. *European Journal of Operation Research*, 164(3):792–799.