Value-Difference Based Exploration: Adaptive Control between Epsilon-Greedy and Softmax

Michel Tokic[1,2] and Günther Palm[1]

[1] Institute of Neural Information Processing, University of Ulm, 89069 Ulm, Germany
[2] Institute of Applied Research, University of Applied Sciences, Ravensburg-Weingarten, 88241 Weingarten, Germany

Abstract. This paper proposes "Value-Difference Based Exploration combined with Softmax action selection" (VDBE-Softmax) as an adaptive exploration/exploitation policy for temporal-difference learning. The advantage of the proposed approach is that exploration actions are only selected in situations when the knowledge about the environment is uncertain, which is indicated by fluctuating values during learning. The method is evaluated in experiments having deterministic rewards and a mixture of both deterministic and stochastic rewards. The results show that a VDBE-Softmax policy can outperform ε-greedy, Softmax and VDBE policies in combination with on- and off-policy learning algorithms such as Q-learning and Sarsa. Furthermore, it is also shown that VDBE-Softmax is more reliable in case of value-function oscillations.

1 Introduction

Balancing the ratio between exploration and exploitation is one of the most challenging tasks in reinforcement learning with great impact on the agent's learning performance. On the one hand, too much exploration prevents the agent from maximizing short-term reward because selected *exploration* actions may yield negative reward from the environment. On the other hand, *exploiting* uncertain environment knowledge prevents from maximizing long-term reward since selected actions may remain suboptimal. This problem is well known as the *dilemma of exploration and exploitation* [1].

A straightforward—and often very successful—approach is to balance exploration/exploitation by the ε-greedy method [2]. With this method, the amount of exploration is globally controlled by a parameter, ε, that determines the randomness in action selections. In contrast to others, one advantage of ε-greedy is the fact that no memorization of exploration specific data is required, such as counters [3] or confidence bounds [4, 5], which makes the method particularly interesting for very large or even continuous state-spaces. Compared to other more complex methods, ε-greedy is often hard to beat [6] and reported to be often the method of first choice as stated by Sutton [7]. In practice, however, a drawback of ε-greedy is that it is unclear which setting of ε leads to good results for a given learning problem. For this reason, the experimenter has to rigorously

J. Bach and S. Edelkamp (Eds.): KI 2011, LNAI 7006, pp. 335–346, 2011.

hand tune ε for obtaining good results, which can be a very time-consuming task in practice depending on the complexity of the target application.

One method that aims at overcoming the above mentioned limitation of ε-greedy is "Value-Difference Based Exploration" (VDBE) [8]. In contrast to pure ε-greedy, VDBE adapts a state-dependent exploration-probability, $\varepsilon(s)$, based on fluctuations in the temporal-difference error instead of requiring to tune a global parameter by hand. However, since the original article on VDBE demonstrated the method on a multi-armed bandit task [9], results from applying the method in multi-state MDPs are still due. For this reason, open questions are: (1) is the method also able to outperform other basic exploration strategies in multi-state MDPs and (2) how do *on-* and *off-policy* learning methods affect learning performance?

This paper gives answers to these questions: Results are reported on evaluating ε-greedy, Softmax and VDBE policies on two different examples. In this context, it is shown that value-function oscillations (e.g. caused by function approximation or by learning algorithms such as Sarsa) can lead to a constant level of exploration when using VDBE and thus to bad learning performance. For this reason, an extension of VDBE to the so-called *VDBE-Softmax* method is propsed that extends Wierings' *Max-Boltzmann Exploration* rule [10] in an adaptive manner. In Section 4, all four policies (ε-greedy, Softmax, VDBE and VDBE-Softmax) are evaluated on the cliff-walking problem [1] using determinstic rewards. In Section 5, all four policies are evaluated in the here presented bandit-world task having both determinstic and stochastic rewards. The results show that VDBE and VDBE-Softmax policies are able to outperform ε-greedy- and Softmax policies under the condition of using Q-learning and constant values for the exploration parameter. Finally, we show that VDBE diverges in case of oscillations in the value function, but in turn converges to near optimal results when the proposed VDBE-Softmax method is used.

2 Methodology

The reinforcement learning (RL) framework is considered where an agent interacts with a Markovian decision process (MDP) [1]. At each discrete time step $t \in \{0, 1, 2, ...\}$ the agent is in a certain state $s_t \in \mathcal{S}$. After the selection of an action, $a_t \in \mathcal{A}(s_t)$, the agent receives a reward signal from the environment, $r_{t+1} \in \mathbb{R}$, and passes into a successor state s'. The decision which action is chosen in a certain state is characterized by a policy $\pi(s) = a$, which could also be stochastic $\pi(a|s) = Pr\{a_t = a|s_t = s\}$. A policy that maximizes the cumulative reward is denoted as π^*.

In reinforcement learning, one way of learning policies is learning a value-function that denotes how "valuable" it is to select action a in state s. Here, a state-action value, $Q(s, a)$, denotes the expected discounted reward for following π when starting in state s and selecting action a:

$$Q^\pi(s, a) = E_\pi \left\{ \sum_{k=0}^{\infty} \gamma^k r_{t+k+1} | s_t = s, a_t = a \right\} \quad , \tag{1}$$

where γ is a discount factor such that $0 < \gamma \leq 1$ for episodic learning tasks and $0 < \gamma < 1$ for continuous learning tasks.

2.1 Learning the *Q* Function by *On*- and *Off*-Policy Methods

Value functions are learned by sampling observations of the interaction between the agent and its environment. For this, the branch of *temporal-difference learning* offers two commonly used algorithms which are namely Sarsa for on-policy control [11]:

$$\begin{aligned} \Delta_{\text{Sarsa}} &\leftarrow [r_{t+1} + \gamma Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)] \\ Q(s_t, a_t) &\leftarrow Q(s_t, a_t) + \alpha \Delta_{\text{Sarsa}} \ , \end{aligned} \tag{2}$$

and Q-learning for off-policy control [2]:

$$\begin{aligned} b^* &\leftarrow \text{argmax}_{b \in \mathcal{A}(s_{t+1})} Q(s_{t+1}, b) \\ \Delta_{\text{Qlearning}} &\leftarrow [r_{t+1} + \gamma Q(s_{t+1}, b^*) - Q(s_t, a_t)] \\ Q(s_t, a_t) &\leftarrow Q(s_t, a_t) + \alpha \Delta_{\text{Qlearning}} \ , \end{aligned} \tag{3}$$

where α is a stepsize parameter [12]. The only technical difference between both algorithms is the inclusion of successor-state information used for the evaluation of action a_t taken in state s_t while learning the value function. Sarsa includes the discounted value of the selected action in the successor state, $Q(s_{t+1}, a_{t+1})$, for which reason it is called to be an *on-policy* method. In contrast, Q-learning includes the discounted value of the optimal action in the successor state, $Q(s_{t+1}, b^*)$, for which reason it is called to be an *off-policy* method.

Although the convergence of Sarsa depends on the stochasticity in action selections, it is well known that the algorithm outperforms Q-learning in many cases even though no convergence proof exists for Sarsa. However, if the stochasticity in action selections becomes zero (i.e. greedy), Sarsa technically becomes the same as Q-learning, and thus also convergent under several conditions [13].

2.2 Basic Exploration/Exploitation Strategies

Two widely used methods for balancing exploration/exploitation are ε-greedy and Softmax [1]. With ε-greedy, at each time step, the agent selects a random action with a fixed probability, $0 \leq \varepsilon \leq 1$, instead of selecting greedily one of the learned optimal actions with respect to the Q-function:

$$\pi(s) = \begin{cases} \text{random action from } \mathcal{A}(s) & \text{if } \xi < \varepsilon \\ \text{argmax}_{a \in \mathcal{A}(s)} Q(s, a) & \text{otherwise,} \end{cases} \tag{4}$$

where $0 \leq \xi \leq 1$ is a uniform random number drawn at each time step. In contrast, Softmax utilizes action-selection probabilities which are determined by ranking the value-function estimates using a Boltzmann distribution:

$$\pi(a|s) = Pr\{a_t = a | s_t = s\} = \frac{e^{\frac{Q(s,a)}{\tau}}}{\sum_b e^{\frac{Q(s,b)}{\tau}}} \ , \tag{5}$$

where τ is a positive parameter called temperature. High temperatures cause all actions to be nearly equiprobable, whereas low temperatures cause greedy action selections.

In practice, both methods have advantages and disadvantages as described in [1]. In the literature, both policies have been reported as methods for describing the action-selection process in the human brain, where Softmax seems to be the better fit of both [14].

2.3 Value-Difference Based Exploration

In order to control the agent's action-selection policy, the basic idea of "Value-Difference Based Exploration" (VDBE) is to extend the ε-greedy method by introducing a state-dependent exploration probability, $\varepsilon(s)$, instead of hand-tuning a global parameter [8]. The desired behavior is to have the agent being more explorative in situations when the knowledge about the environment is uncertain, e.g. at the beginning of the learning process, which is indicated by fluctuating values during learning. On the other hand, the amount of exploration should be reduced as far as the agent's knowledge becomes certain, which is indicated by very small or no value differences. Such an adaptive behavior is obtained by computing after each learning step a state-dependent exploration probability, $\varepsilon(s)$, according to the difference in a Boltzmann distribution of the value before and after learning:

$$
\begin{aligned}
f(s,a,\sigma) &= \left| \frac{e^{\frac{Q_t(s,a)}{\sigma}}}{e^{\frac{Q_t(s,a)}{\sigma}} + e^{\frac{Q_{t+1}(s,a)}{\sigma}}} - \frac{e^{\frac{Q_{t+1}(s,a)}{\sigma}}}{e^{\frac{Q_t(s,a)}{\sigma}} + e^{\frac{Q_{t+1}(s,a)}{\sigma}}} \right| \\
&= \frac{1 - e^{\frac{-|Q_{t+1}(s,a) - Q_t(s,a)|}{\sigma}}}{1 + e^{\frac{-|Q_{t+1}(s,a) - Q_t(s,a)|}{\sigma}}} = \frac{1 - e^{\frac{-|\alpha \cdot \Delta|}{\sigma}}}{1 + e^{\frac{-|\alpha \cdot \Delta|}{\sigma}}} \qquad (6) \\
\varepsilon_{t+1}(s) &= \delta \cdot f(s_t, a_t, \sigma) + (1 - \delta) \cdot \varepsilon_t(s) \ , \qquad (7)
\end{aligned}
$$

where σ is a positive constant called *inverse sensitivity* and $\delta \in [0, 1)$ a parameter determining the influence of the selected action on the state-dependent exploration probability. A reasonable setting for δ is the inverse of the number of actions in the current state, $\delta(s) = \frac{1}{|\mathcal{A}(s)|}$, since all actions should contribute equally to $\varepsilon(s)$, and which always led to good results in our experiments. At the beginning of the learning process, all exploration probabilities are initialized arbitrary, e.g. $\varepsilon_{t=0}(s) = 1$ for all states. The parameter σ influences $\varepsilon(s)$ in a way that low values cause full exploration at small value changes. On the other hand, high values of σ cause a high level of exploration only at large value changes. Finally, the exploration probability approaches zero as far as the Q-function converges which results to pure greedy action selections.

3 VDBE-Softmax

Although VDBE has successfully been applied in solving bandit problems with stationary reward distributions [8], one drawback of VDBE (in particular of ε-greedy) is that exploration actions are chosen uniformly distributed among all

possible actions in the current state. Such exploration behavior can lead to bad performance when many actions in the current state yield to relatively high negative reward, even if this knowledge is present through already learned Q values. Furthermore, Q-function oscillations cause a non-zero level of $\varepsilon(s)$, e.g. caused by stochastic rewards or by function approximators for the Q-function. In turn, this causes excessive selections of bad actions in cases when only a few actions lead to positive reward.

A way of relaxing the above drawback is by combining an ε-greedy policy with Softmax (Equation 5) as proposed by Wiering as the *Max-Boltzmann Exploration* method (MBE) [10]. MBE behaves the same as ε-greedy except that exploration actions are selected according to the Softmax rule:

$$\pi(s) = \begin{cases} \text{Softmax action according to Equation 5} & \text{if } \xi < \varepsilon \\ \text{argmax}_{a \in \mathcal{A}(s)} Q(s,a) & \text{otherwise,} \end{cases} \tag{8}$$

where ξ is a uniform random number from the interval $[0, 1]$. Although *Max-Boltzmann Exploration* requires two parameters to be set (τ and ε), advantages of both methods are combined into one method [10].

The idea of combining both methods is now used for extending VDBE to the so-called VDBE-Softmax method. In contrast to MBE, VDBE-Softmax adapts the state-dependent exploration rate $\varepsilon(s)$ according to VDBE but selects random actions according to Softmax in case of $\xi < \varepsilon(s)$. Furthermore, in order to ease the search for reasonable parameters for Softmax, we propose using a normalization of the Q values into the interval $[V_{\text{normMin}}, V_{\text{normMax}}]$, e.g. $[-1, 1]$, and having the temperature parameter of Softmax set constantly to the value of $\tau = 1$. With this, a mean independency of the distribution of Q values in state s is achieved that enables the selection of τ more intuitively. In our experiments, such an approach turned out to be sufficient for suppressing the selection of actions yielding to highly negative reward in case of $\xi < \varepsilon(s)$. Finally, Algorithm 1 depicts the interaction between Q-learning and VDBE-Softmax, where SARSA is combined analogously when replacing lines 13-15 of Algorithm 1 according to Equation 2.

4 Experiments in the Cliff-Walking Task

The proposed method has been evaluated on the cliff-walking task presented by Sutton and Barto [1]. As shown in Figure 1, the goal of the agent is to learn a path from the start state, S, to the goal state, G. For each step, the agent receives a reward of $r = -1$ except for falling off the cliff which is rewarded by $r = -100$, and where the agent is instantly sent back to S.

In the cliff-walking task, Sutton and Barto demonstrated the different learning behaviors of *on-* and *off*-policy methods when using stochastic policies[1]. As a result, the agent learns the safe path when using Sarsa, but the optimal path when using Q-learning. The optimal (shortest) path, however, is a bad choice

[1] In particular, Sutton and Barto used ε-greedy having $\varepsilon = 0.1$.

Algorithm 1. Q-LEARNING WITH VDBE-SOFTMAX

Initialize $Q(s,a)$ arbitrarily, e.g. $Q(s,a) = 0$ for all s, a
Initialize $\varepsilon(s)$ arbitrarily, e.g. $\varepsilon(s) = 1$ for all s
for each episode **do**
  Initialize start state s
  repeat
    $\xi \leftarrow \text{rand}(0..1)$
    if $\xi < \varepsilon(s)$ **then**
      $a \leftarrow \text{SOFTMAX}(\mathcal{A}(s))$
    else
      $a \leftarrow \text{argmax}_{b \in \mathcal{A}(s)} Q(s,b)$
    end if
    take action a, observe reward r and successor state s'
    $b^* \leftarrow \text{argmax}_{b \in \mathcal{A}(s')} Q(s',b)$
    $\Delta \leftarrow r + \gamma Q(s',b^*) - Q(s,a)$
    $Q(s,a) \leftarrow Q(s,a) + \alpha\Delta$
    $\varepsilon(s) \leftarrow \delta \cdot \frac{1-e^{\frac{-|\alpha \cdot \Delta|}{\sigma}}}{1+e^{\frac{-|\alpha \cdot \Delta|}{\sigma}}} + (1-\delta) \cdot \varepsilon(s)$
    $s \leftarrow s'$
  until s is terminal state
end for

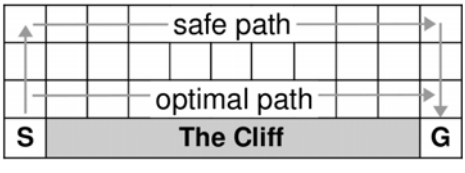

Fig. 1. The cliff-walking task as presented by Sutton and Barto [1]

in this example since the agent will travel right along the edge of the cliff and which occasionally results in falling off.

4.1 Experiment Setup

The cliff-walking experiment has been setup as follows and having the results averaged over 1000 experiments each having 400 episodes. An episode begins in the start state, S, and terminates when: (1) the agent has walked a maximum of 100 steps, or (2) the agent arrived at the goal state, G. Throughout the experiment, the step-size parameter α has been constantly set to the value of $\alpha = 0.8$.

Since the learning problem is an episodic task, no discounting ($\gamma = 1$) has been used. In this experimental setting, Q-learning and Sarsa have been investigated with ε-greedy, Softmax and VDBE policies using constant parameter settings, and using a tabular approximation of the value function. In experiments with VDBE and VDBE-Softmax, all exploration probabilities have been initialized with $\varepsilon(s) = 1$, as well all δ's been configured with $\delta(s) = \frac{1}{|\mathcal{A}(s)|}$. For VDBE-Softmax, the normalization interval has been set to $[0, 1]$ using $\tau = 1$ for the Softmax method. At the beginning of each experiment, all state-action values have been optimistically initialized with $Q_{t=0}(s, a) = 0$, thus causing additional exploration in the first phase of learning.

4.2 Results

The experimental results of the cliff-walking study are shown in Figure 2. It is observable that all four exploration methods perform optimal when having (almost) a greedy exploration parameter configured, i.e. $\varepsilon = 0$ for ε-greedy, $\tau = 0.04$ for Softmax, or $\sigma = 100$ for VDBE and VDBE-Softmax. In case of stochastic policies, it can be observed that Sarsa outperforms Q-learning when using ε-greedy or Softmax policies, which confirms the results of Sutton and Barto. Interestingly, the performance of Q-learning in conjunction with ε-greedy is sometimes even better in the first episodes compared to the converged performance in the last episodes, e.g. when ε is set to 0.5. The effect of unlearning such an apparently better behavior is caused by greedy action selections in the first phase of learning, and also led back to the insecureness of the value-function estimates. Due to this, the agent walks more often away from the cliff during the first episodes, but travels right along the edge once the value function converges to the true values. In terms of learning speed, no remarkable changes other than the speed of learning were observable for different settings of α when using ε-greedy or Softmax.

In contrast, a different behavior is observable for VDBE as shown in Figure 2(c). Due to the nature of pursuing to greedy, VDBE in conjunction with Q-learning always converged to the optimal results under any settings of σ. On the contrary, VDBE in combination with Sarsa shows to converge to the optimal results only for high inverse sensitivities ($\sigma \gtrsim 10$), but diverges for values of $\sigma < 10$. The reason for this behavior is that Sarsa has no convergence guarantee, and oscillations in the value function are caused when the policy is stochastic. Furthermore, these oscillations cause VDBE to increase the exploration probability, thus causing the agent to explore its environment constantly. In evaluations for other values of α, the σ parameter could be more reduced the more α is reduced at the same time. The advantage of additionally combining Softmax to the VDBE-Softmax method is shown in Figure 2(d). In contrast to VDBE, the results of Sarsa in conjunction with VDBE-Softmax also converge to near optimal results independently of σ, which only had influence on the speed of convergence.

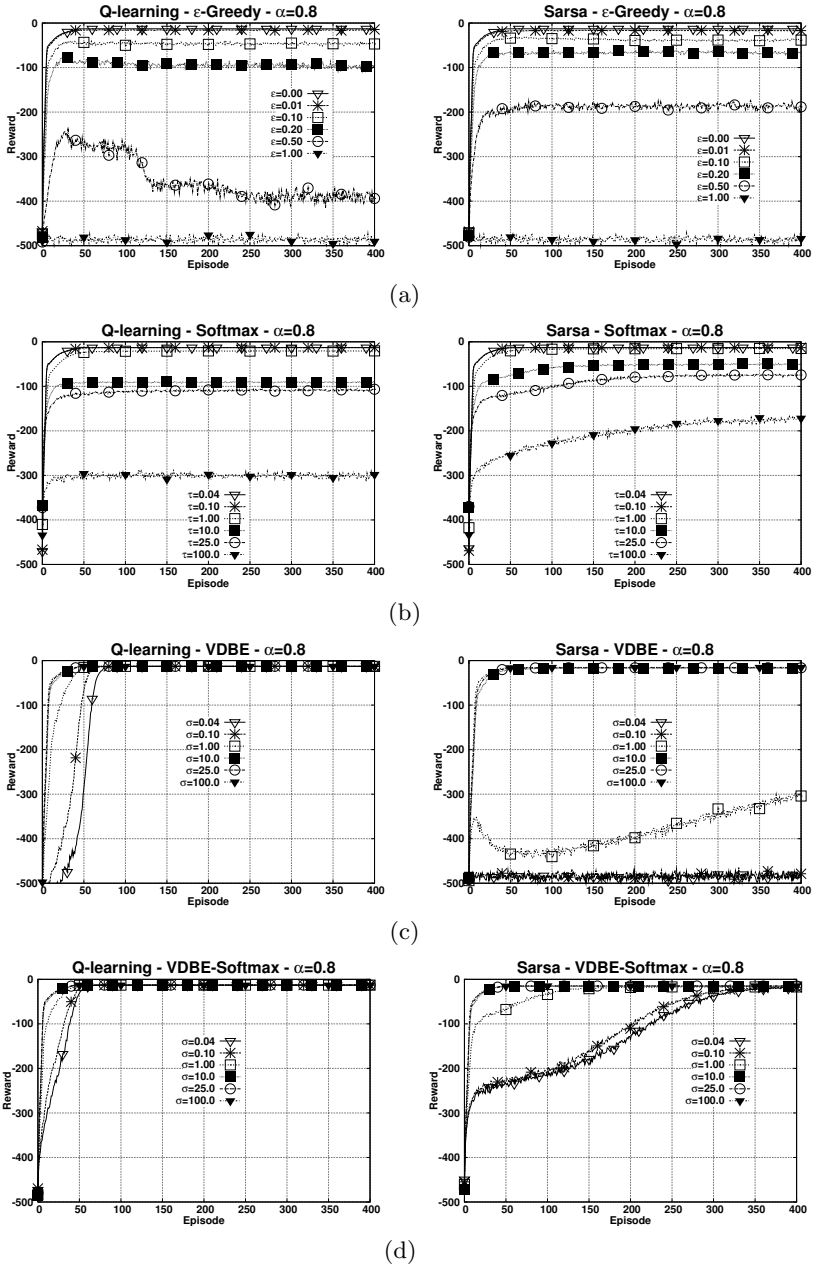

Fig. 2. Comparison of the cumulative reward per episode on the cliff-walking task using Sarsa and Q-learning in conjunction with: (a) ε-greedy, (b) Softmax, (c) VDBE and (d) VDBE-Softmax. Results are averaged over 1000 experiments.

5 Experiments in the Bandit-World Task

The second experiment has been conducted in an extension of the multi-armed bandit problem proposed here as the *"Bandit-World"* problem. In addition to the original multi-armed bandit problem [9], the environment in the bandit world consists of multiple states and diverse bandits (in this example states $\mathbf{B_1}$ and $\mathbf{B_2}$). The reward distributions of bandit states are unequal and the agent has to decide whether it sticks with the first bandit it reaches or whether it travels around in hope to find another (maybe better) bandit. Traveling around is expensive since each transition to another state is rewarded negatively by the value of $r = -1$. On the contrary, the reward for choosing a bandit lever in states $\mathbf{B_1}$ and $\mathbf{B_2}$ is drawn randomly according to a normal distribution $\mathcal{N}(Q^*(\mathbf{B}, a_{\text{lever}}), 1)$.

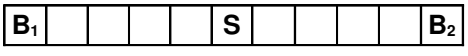

Fig. 3. The bandit-world task. **S** indicates the start state; $\mathbf{B_1}$ and $\mathbf{B_2}$ indicate two different bandits.

5.1 Experiment Setup

The bandit world has been evaluated as follows. Since the environment is an episodic multi-state MDP having the rewards be a mixture of both deterministic and stochastic scalars, we evaluated low values for the step-size parameter α. Each bandit consists of three levers with mean values $Q^*(\mathbf{B_1}) = \{-1.0, 0.0, 1.0\}$ and $Q^*(\mathbf{B_2}) = \{2.0, 3.0, 4.0\}$. In bandit states, a fourth possible action is allowed (a_{left} or a_{right}) that leaves the bandit and which is rewarded by $r = -1$ (which also applies for every other transition in non-bandit states). Results are averaged over 500 experiments each running over 400 episodes. An episode begins in the start state **S** and terminates after $T = 100$ steps. In this experimental setting, Q-learning and Sarsa have been investigated with ε-greedy, Softmax and VDBE policies using constant parameter settings, and using a tabular approximation of the value function. In experiments with VDBE and VDBE-Softmax, all exploration probabilities have been initialized with $\varepsilon(s) = 1$, as well all δ's been configured with $\delta(s) = \frac{1}{|\mathcal{A}(s)|}$. All state-action values have been optimistically initialized with $Q_{t=0}(s, a) = 0$, thus causing additional exploration in the first phase of learning. Furthermore, the Softmax method used by VDBE-Softmax has been configured with temperature $\tau = 1$ and normalization boundaries $[-1, 1]$. Finally, we evaluated discounting in the bandit-world task with $\gamma = 0.9$.

5.2 Results

Figure 4 shows the results of the bandit-world experiments. For ε-greedy and Softmax policies, almost no performance difference is observable when using Q-learning and Sarsa. For both policies, the worst-case reward/episode is about -50 in case the action-selection policy is pure random ($\varepsilon = 1.0$ and $\tau \geq 100.0$).

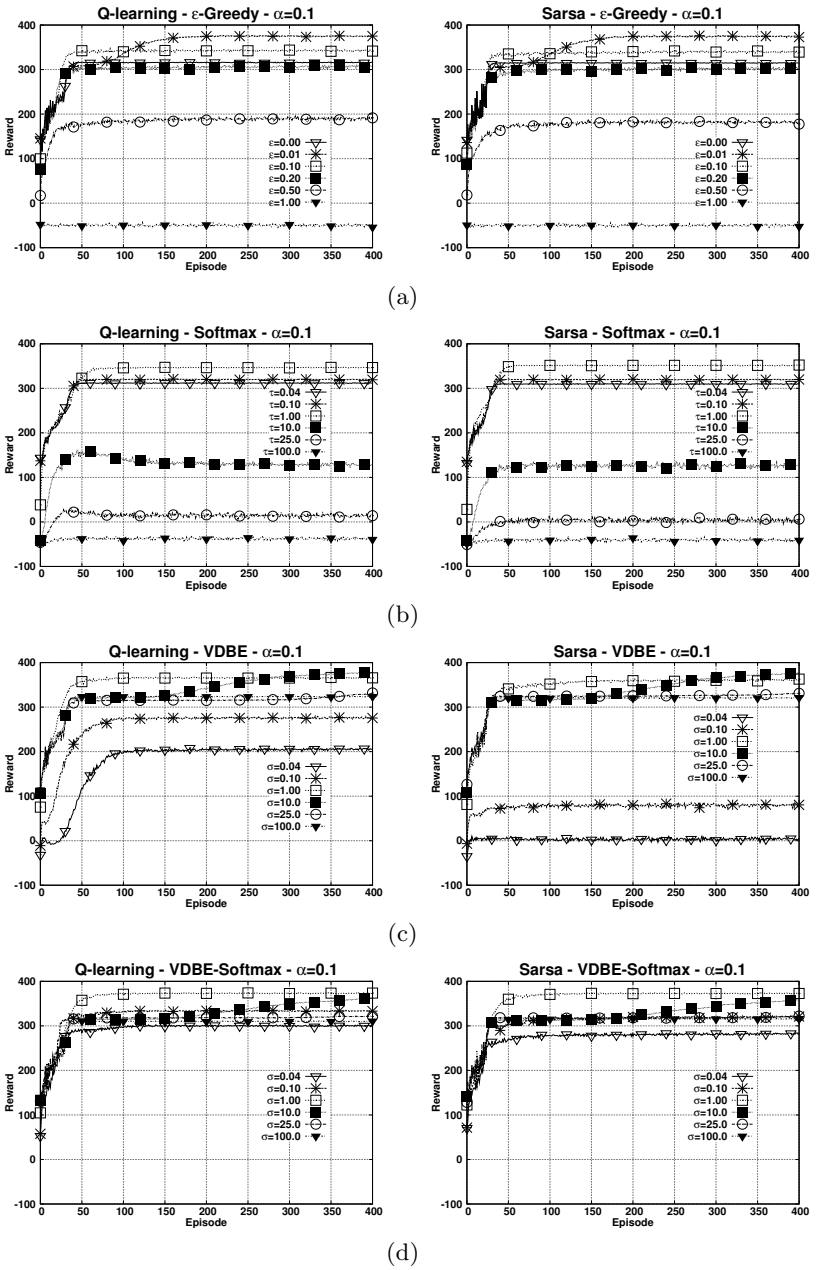

Fig. 4. Comparison of the cumulative reward per episode on the bandit-world task using Sarsa and Q-learning in conjunction with: (a) ε-greedy, (b) Softmax, (c) VDBE and (d) VDBE-Softmax. Results are averaged over 500 experiments.

In contrast, VDBE in combination with Q-learning converged to a worst-case reward/episode of about 200, and to 300 for VDBE-Softmax respectively. In combination with Sarsa, VDBE shows (again) to diverge when using low values for σ, since Sarsa causes value-function oscillations in case of stochastic policies. In turn, VDBE converged to near optimal results for settings of $\sigma \geq 1$. Finally, the results show that the worst-case reward/episode is much higher for VDBE-Softmax compared to the other three methods. Interestingly, VDBE, VDBE-Softmax and Softmax policies turned out to maximize the cumulative reward when setting the exploration parameter (σ or τ) to the value of 1.

6 Discussions and Conclusions

This paper showed that VDBE can successfully be applied to balance exploration/exploitation also in multi-state MDPs, which answers the first open question mentioned above. The obtained results lead to the conclusion that VDBE in conjunction with Q-learning is able to outperform other basic exploration methods, since the information is not only based on current information of the value function, but also biased on the progress of learning the function. The results highlight the importance of using learning algorithms that are proven to converge. As a counterexample, the experiment with Sarsa in conjunction with VDBE revealed that oscillations in the value function can lead to a constant level of exploration, thus to full exploration in the worst-case. Such behavior can sometimes successfully be handled by: (1) a fine-tuning of the inverse sensitivity σ, (2) a fine-tuning of the step-size parameter α or (3) by using the proposed VDBE-Softmax method. Finally, this fact answers the second open question because convergence proofs exist for Q-learning (*off-policy* control) rather than for Sarsa (*on-policy* control).

Although results were optimal using a pure greedy policy in the cliff-walking task, this setting does not apply for every learning problem as well, and most often a bit of exploration improves learning performance as shown in the bandit-world example. In fact, a pure greedy policy is most often sub-optimal, and less examples such as the cliff-walking problem exist that show the contrary. Only in the limit, the policy should converge to greedy as far as enough information about the environment has been sampled, and which has also been shown in a preceding study of VDBE on the multi-armed bandit problem [8]. Finally, the results also show that extending VDBE with the Softmax method converged much more reliable to near optimal results under a wide range of parameter configurations.

To sum up, the presented results suggest that balancing exploration and exploitation on basis of fluctuations in the value function is a reasonable technique for supporting the decision-making process in reinforcement learning.

Acknowledgements. The authors like to thank F. Schwenker[1], W. Ertel[2], P. Ertle[2] and R. Cubek[2] for valuable comments and discussions.

This work has been conducted within the Collaborative Center for Applied Research on Service Robotics (ZAFH Service Robotics). The authors gratefully acknowledge the research grants of the state Baden-Württemberg and the European Union.

References

[1] Sutton, R.S., Barto, A.G.: Reinforcement Learning: An Introduction. MIT Press, Cambridge (1998)
[2] Watkins, C.: Learning from Delayed Rewards. PhD thesis, University of Cambridge, Cambridge, England (1989)
[3] Thrun, S.B.: Efficient exploration in reinforcement learning. Technical Report CMU-CS-92-102, Carnegie Mellon University, Pittsburgh, PA, USA (1992)
[4] Kaelbling, L.P.: Learning in embedded systems. MIT Press, Cambridge (1993)
[5] Auer, P.: Using confidence bounds for exploitation-exploration trade-offs. Journal of Machine Learning Research 3, 397–422 (2002)
[6] Vermorel, J., Mohri, M.: Multi-armed bandit algorithms and empirical evaluation. In: Gama, J., Camacho, R., Brazdil, P., Jorge, A., Torgo, L. (eds.) ECML 2005. LNCS (LNAI), vol. 3720, pp. 437–448. Springer, Heidelberg (2005)
[7] Heidrich-Meisner, V.: Interview with Richard S. Sutton. In: Künstliche Intelligenz, vol. 3, pp. 41–43 (2009)
[8] Tokic, M.: Adaptive ε-greedy exploration in reinforcement learning based on value differences. In: Dillmann, R., Beyerer, J., Hanebeck, U.D., Schultz, T. (eds.) KI 2010. LNCS, vol. 6359, pp. 203–210. Springer, Heidelberg (2010)
[9] Robbins, H.: Some aspects of the sequential design of experiments. Bulletin of the American Mathematical Society 58, 527–535 (1952)
[10] Wiering, M.: Explorations in Efficient Reinforcement Learning. PhD thesis, University of Amserdam, Amsterdam (1999)
[11] Rummery, G.A., Niranjan, M.: On-line Q-learning using connectionist systems. Technical Report CUED/F-INFENG/TR 166, Cambridge University (1994)
[12] George, A.P., Powell, W.B.: Adaptive stepsizes for recursive estimation with applications in approximate dynamic programming. Machine Learning 65(1), 167–198 (2006)
[13] Watkins, C., Dayan, P.: Technical note: Q-learning. Machine Learning 8(3), 279–292 (1992)
[14] Daw, N.D., O'Doherty, J.P., Dayan, P., Seymour, B., Dolan, R.J.: Cortical substrates for exploratory decisions in humans. Nature 441, 876–879 (2006)