Characterizing Reinforcement Learning Methods through Parameterized Learning Problems

Shivaram Kalyanakrishnan and Peter Stone

Department of Computer Science, The University of Texas at Austin

August 2011

Practice $\Longrightarrow$ Imperfect Representation

Task	**State Aliasing**	**State Space**	**Policy Representation (Number of features)**
Backgammon (T1992)			
Job-shop scheduling (ZD1995)			
Elevator dispatching (CB1996)			
Acrobot control (S1996)			
Dynamic channel allocation (SB1997)			
Active guidance of finless rocket (GM2003)			
Robot sensing strategy (KF2004)			
Helicopter control (NKJS2004)			
Adaptive job routing/scheduling (WS2004)			
Robot soccer keepaway (SSK2005)			
Robot obstacle negotiation (LSYSN2006)			
Tetris (SL2006)			
Optimized trade execution (NFK2007)			
9×9 Go (SSM2007)			
Autonomic resource allocation (TJDB2007)			
General game playing (FB2008)			
Soccer opponent “hassling” (GRT2009)			
Adaptive epilepsy treatment (GVAP2008)			
Computer memory scheduling (IMMC2008)			
Motor skills (PS2008)			

Practice $\Longrightarrow$ Imperfect Representation

Task	**State Aliasing**	**State Space**	**Policy Representation (Number of features)**
Backgammon (T1992)	Absent	Discrete	Neural network (198)
Job-shop scheduling (ZD1995)	Absent	Discrete	Neural network (20)
Elevator dispatching (CB1996)	Present	Continuous	Neural network (46)
Acrobot control (S1996)	Absent	Continuous	Tile coding (4)
Dynamic channel allocation (SB1997)	Absent	Discrete	Linear (100's)
Active guidance of finless rocket (GM2003)	Present	Continuous	Neural network (14)
Robot sensing strategy (KF2004)	Present	Continuous	Linear (36)
Helicopter control (NKJS2004)	Present	Continuous	Neural network (10)
Adaptive job routing/scheduling (WS2004)	Present	Discrete	Tabular (4)
Robot soccer keepaway (SSK2005)	Present	Continuous	Tile coding (13)
Robot obstacle negotiation (LSYSN2006)	Present	Continuous	Linear (10)
Tetris (SL2006)	Absent	Discrete	Linear (22)
Optimized trade execution (NFK2007)	Present	Discrete	Tabular (2-5)
9 $\times$ 9 Go (SSM2007)	Absent	Discrete	Linear ($\approx$1.5 million)
Autonomic resource allocation (TJDB2007)	Present	Continuous	Neural network (2)
General game playing (FB2008)	Absent	Discrete	Tabular (part of state space)
Soccer opponent "hassling" (GRT2009)	Present	Continuous	Neural network (9)
Adaptive epilepsy treatment (GVAP2008)	Present	Continuous	Extremely rand. trees (114)
Computer memory scheduling (IMMC2008)	Absent	Discrete	Tile coding (6)
Motor skills (PS2008)	Present	Continuous	Motor primitive coeff. (100's)

Practice $\Longrightarrow$ Imperfect Representation

Task	State Aliasing	State Space	Policy Representation (Number of features)
Backgammon (T1992)	Absent	Discrete	Neural network (198)
Job-shop scheduling (ZD1995)	Absent	Discrete	Neural network (20)
Elevator dispatching (CB1996)	Present	Continuous	Neural network (46)
Acrobot control (S1996)	Absent	Continuous	Tile coding (4)
Dynamic channel allocation (SB1997)	Absent	Discrete	Linear (100's)
Active guidance of finless rocket (GM2003)	Present	Continuous	Neural network (14)
Robot sensing strategy (KF2004)	Present	Continuous	Linear (36)
Helicopter control (NKJS2004)	Present	Continuous	Neural network (10)
Adaptive job routing/scheduling (WS2004)	Present	Discrete	Tabular (4)
Robot soccer keepaway (SSK2005)	Present	Continuous	Tile coding (13)
Robot obstacle negotiation (LSYSN2006)	Present	Continuous	Linear (10)
Tetris (SL2006)	Absent	Discrete	Linear (22)
Optimized trade execution (NFK2007)	Present	Discrete	Tabular (2-5)
9×9 Go (SSM2007)	Absent	Discrete	Linear ($\approx$1.5 million)
Autonomic resource allocation (TJDB2007)	Present	Continuous	Neural network (2)
General game playing (FB2008)	Absent	Discrete	Tabular (part of state space)
Soccer opponent "hassling" (GRT2009)	Present	Continuous	Neural network (9)
Adaptive epilepsy treatment (GVAP2008)	Present	Continuous	Extremely rand. trees (114)
Computer memory scheduling (IMMC2008)	Absent	Discrete	Tile coding (6)
Motor skills (PS2008)	Present	Continuous	Motor primitive coeff. (100's)

Practice $\Longrightarrow$ Imperfect Representation

Task	State Aliasing	State Space	Policy Representation (Number of features)
Backgammon (T1992)	Absent	Discrete	Neural network (198)
Job-shop scheduling (ZD1995)	Absent	Discrete	Neural network (20)
Elevator dispatching (CB1996)	Present	Continuous	Neural network (46)
Acrobot control (S1996)	Absent	Continuous	Tile coding (4)
Dynamic channel allocation (SB1997)	Absent	Discrete	Linear (100's)
Active guidance of finless rocket (GM2003)	Present	Continuous	Neural network (14)
Robot sensing strategy (KF2004)	Present	Continuous	Linear (36)
Helicopter control (NKJS2004)	Present	Continuous	Neural network (10)
Adaptive job routing/scheduling (WS2004)	Present	Discrete	Tabular (4)
Robot soccer keepaway (SSK2005)	Present	Continuous	Tile coding (13)
Robot obstacle negotiation (LSYSN2006)	Present	Continuous	Linear (10)
Tetris (SL2006)	Absent	Discrete	Linear (22)
Optimized trade execution (NFK2007)	Present	Discrete	Tabular (2-5)
9 $\times$ 9 Go (SSM2007)	Absent	Discrete	Linear ($\approx$1.5 million)
Autonomic resource allocation (TJDB2007)	Present	Continuous	Neural network (2)
General game playing (FB2008)	Absent	Discrete	Tabular (part of state space)
Soccer opponent "hassling" (GRT2009)	Present	Continuous	Neural network (9)
Adaptive epilepsy treatment (GVAP2008)	Present	Continuous	Extremely rand. trees (114)
Computer memory scheduling (IMMC2008)	Absent	Discrete	Tile coding (6)
Motor skills (PS2008)	Present	Continuous	Motor primitive coeff. (100's)

Practice $\Longrightarrow$ Imperfect Representation

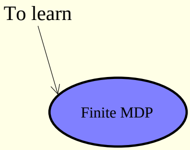
Task	State Aliasing	State Space	Policy Representation (Number of features)
Backgammon (T1992)	Absent	Discrete	Neural network (198)
Job-shop scheduling (ZD1995)	Absent	Discrete	Neural network (20)
Elevator dispatching (CB1996)	Present	Continuous	Neural network (46)
Acrobot control (S1996)	Absent	Continuous	Tile coding (4)
Dynamic channel allocation (SB1997)	Absent	Discrete	Linear (100's)
Active guidance of finless rocket (GM2003)	Present	Continuous	Neural network (14)
Robot sensing strategy (KF2004)	Present	Continuous	Linear (36)
Helicopter control (NKJS2004)	Present	Continuous	Neural network (10)
Adaptive job routing/scheduling (WS2004)	Present	Discrete	Tabular (4)
Robot soccer keepaway (SSK2005)	Present	Continuous	Tile coding (13)
Robot obstacle negotiation (LSYSN2006)	Present	Continuous	Linear (10)
Tetris (SL2006)	Absent	Discrete	Linear (22)
Optimized trade execution (NFK2007)	Present	Discrete	Tabular (2-5)
9 $\times$ 9 Go (SSM2007)	Absent	Discrete	Linear ($\approx$1.5 million)
Autonomic resource allocation (TJDB2007)	Present	Continuous	Neural network (2)
General game playing (FB2008)	Absent	Discrete	Tabular (part of state space)
Soccer opponent "hassling" (GRT2009)	Present	Continuous	Neural network (9)
Adaptive epilepsy treatment (GVAP2008)	Present	Continuous	Extremely rand. trees (114)
Computer memory scheduling (IMMC2008)	Absent	Discrete	Tile coding (6)
Motor skills (PS2008)	Present	Continuous	Motor primitive coeff. (100's)

Practice $\Longrightarrow$ Imperfect Representation

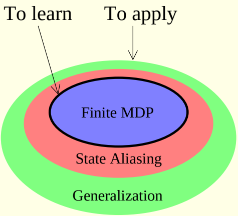
Task	State Aliasing	State Space	Policy Representation (Number of features)
Backgammon (T1992)	Absent	Discrete	Neural network (198)
Job-shop scheduling (ZD1995)	Absent	Discrete	Neural network (20)
Elevator dispatching (CB1996)	Present	Continuous	Neural network (46)
Acrobot control (S1996)	Absent	Continuous	Tile coding (4)
Dynamic channel allocation (SB1997)	Absent	Discrete	Linear (100's)
Active guidance of finless rocket (GM2003)	Present	Continuous	Neural network (14)
Robot sensing strategy (KF2004)	Present	Continuous	Linear (36)
Helicopter control (NKJS2004)	Present	Continuous	Neural network (10)
Adaptive job routing/scheduling (WS2004)	Present	Discrete	Tabular (4)
Robot soccer keepaway (SSK2005)	Present	Continuous	Tile coding (13)
Robot obstacle negotiation (LSYSN2006)	Present	Continuous	Linear (10)
Tetris (SL2006)	Absent	Discrete	Linear (22)
Optimized trade execution (NFK2007)	Present	Discrete	Tabular (2-5)
9 $\times$ 9 Go (SSM2007)	Absent	Discrete	Linear ($\approx$1.5 million)
Autonomic resource allocation (TJDB2007)	Present	Continuous	Neural network (2)
General game playing (FB2008)	Absent	Discrete	Tabular (part of state space)
Soccer opponent "hassling" (GRT2009)	Present	Continuous	Neural network (9)
Adaptive epilepsy treatment (GVAP2008)	Present	Continuous	Extremely rand. trees (114)
Computer memory scheduling (IMMC2008)	Absent	Discrete	Tile coding (6)
Motor skills (PS2008)	Present	Continuous	Motor primitive coeff. (100's)

Perfect representations (fully observable, enumerable states) are impractical.

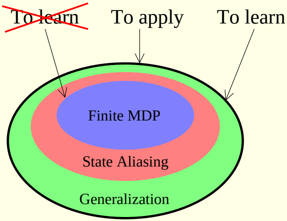
Introspection and Direction



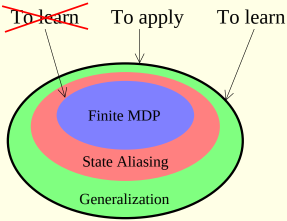
Introspection and Direction



Introspection and Direction



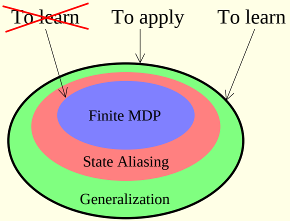
Introspection and Direction



Thesis Question:

"How well do different reinforcement learning methods perform in the presence of state aliasing and function approximation; is it possible to develop methods that are both sample efficient and capable of achieving high asymptotic performance in their presence?"

Introspection and Direction



Thesis Question:

"How well do different reinforcement learning methods perform in the presence of state aliasing and function approximation; is it possible to develop methods that are both sample efficient and capable of achieving high asymptotic performance in their presence?"

Dissertation

Stage 1: Determine which learning methods suit which problems.

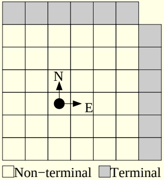
- Parameterized learning problem:
  - s (Size of state space)
  - p (Stochasticity in transitions)
  - χ (“Expressiveness” of function approximator)
  - w (Generalization width)
  - σ (State noise)
- Sarsa $\sim$ VF; CMA-ES $\sim$ PS.

Stage 2: Integrate strengths of different learning methods.

Parameterized Learning Problem: Merits

1. The designed task and learning framework are easy to understand and can be controlled precisely.
2. We may examine the effect of subsets of problem parameters while keeping others fixed.
3. We can benchmark learned policies against optimal behavior.
4. The learning process can be executed in a relatively short duration of time, thereby facilitating extensive experimentation.

Problem Size and Stochasticity



(a)

.77	.33	.59	.34	.82	.39	
.79	.01	.26	.03	.14	.55	.57
.84	.03	.44	.96	.83	.53	.23
.99	.55	.75	.69	.05	.23	.05
.58	.96	.36	.18	.85	.92	.82
.30	.69	.84	.05	.23	.95	.22
	.35	.28	.32	.90	.45	.64

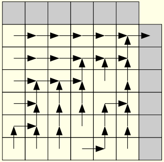
Example Rewards

(b)

0	0	0	0	0	0	
1.2	1.3	1.1	1.1	1.1	0.6	0
3.1	3.2	3.0	2.2	1.5	1.0	0
4.8	4.4	3.7	3.1	2.3	1.4	0
5.9	4.9	4.4	3.7	2.5	1.5	0
6.5	5.8	4.7	3.8	3.3	2.2	0
6.8	6.4	5.5	4.4	3.5	2.9	0

Values of optimal actions

(c)

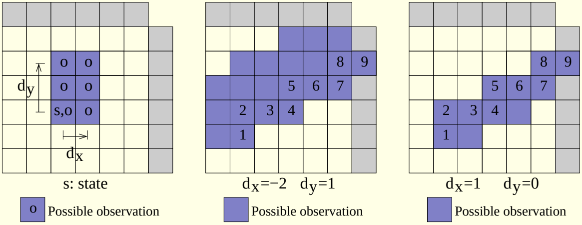


Optimal Policy (p = 0.1)

(d)

Figure: (a) Example of parameterized MDP example with $s = 7$; the number of non-terminal states is 36. (b) Rewards obtained at “next states” of transitions. (c) Optimal action values from each state when $p = 0.1$. (d) Corresponding optimal policy.

State Aliasing



(a) (b) (c)

Figure: An implementation of state aliasing in the example MDP from Figure 1. (a) Variables d_x and d_y (themselves generated randomly based on parameter σ) define a rectangle with the true state at a corner; cells within this rectangle are picked uniformly at random to constitute observed states. (b) A trajectory of true states 1 through 9, and the set of all possible observed states that could be encountered during this trajectory when $d_x = -2$ and $d_y = 1$. (c) For the same trajectory, the set of possible observed states when $d_x = 1$ and $d_y = 0$.

Generalization

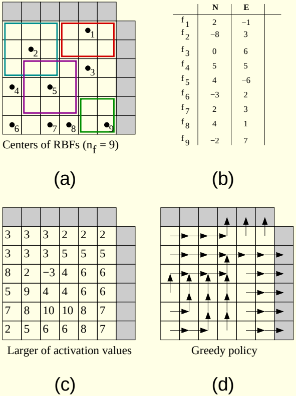


Figure: Generalization scheme in example MDP from Figure 1. (a) A randomly chosen subset of cells (numbered 1 through 9) are the centers of overlapping tiles (giving $\chi = \frac{9}{36} = 0.25$). The tile width w is set to 3; tiles 1, 2, 5, and 9 are shown outlined (and clipped at the boundaries of the non-terminal region). (b) Table showing coefficients associated with each tile for actions **N** and **E**. (c) The activation value of each cell for an action is the sum of the weights of the tiles to which it belongs. The figure shows the higher activation value (among **N** and **E**) for each cell. (d) Arrows mark a policy that is greedy with respect to the activations: that is, in each cell, the action with a higher activation value is chosen.

Parameters of Learning Problem

Table: Summary of learning problem parameters. The last column shows the ranges over which each parameter is valid and meaningful to test.

Parameter	Property of:	Controls:	Range
s	Task	Size of state space	$\{2, 3, \ldots, \infty\}$
p	Task	Stochasticity of transitions	$[0, 0.5)$
σ	Agent/task interface	State aliasing	$[0, \infty)$
χ	Agent	Expressiveness of generalization scheme	$(0, 1]$
w	Agent	Width of generalization	$\{1, 3, \ldots, 2s - 3\}$

On-line Value Function-based (VF) Methods

Sarsa(λ)

ExpSarsa(λ)

Q-learning(λ)

Table: Summary of parameters used by methods within VF. The last column shows the ranges over which we tune each parameter.

Parameter	Controls:	Range
λ	Eligibility traces	$[0, 1]$
α_0	Initial learning rate	$[0.1, 1]$
ϵ_0	Initial exploration rate	$[0.1, 1]$
θ_0	Initial weights	$[-10.0, 10.0]$

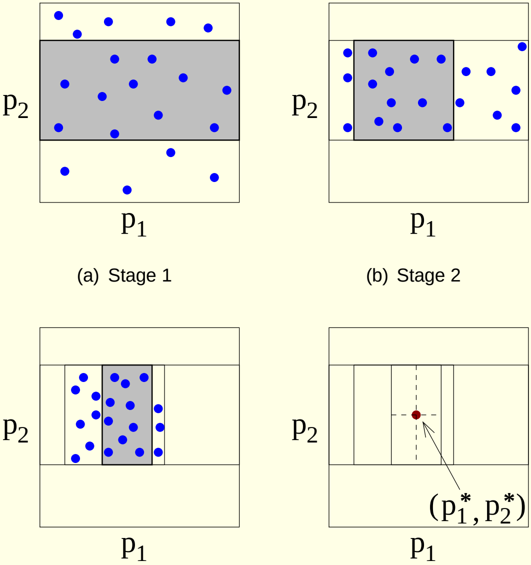
Policy Search (PS) Methods

- Cross-entropy Method (CEM)
- Covariance Matrix Adapation Evolutionary Strategy (CMA-ES)
- Genetic Algorithm (GA)
- Random Weight Guessing (RWG)

Table: Summary of parameters used by methods from PS. The last column shows the ranges over which we tune each parameter. The range shown for $\#$*trials* is used when the total number of episodes is 50,000, as in a majority of our experiments. The range is scaled proportionately with the total number of training episodes. Under RWG, $\#$*trials* is the only method-specific parameter.

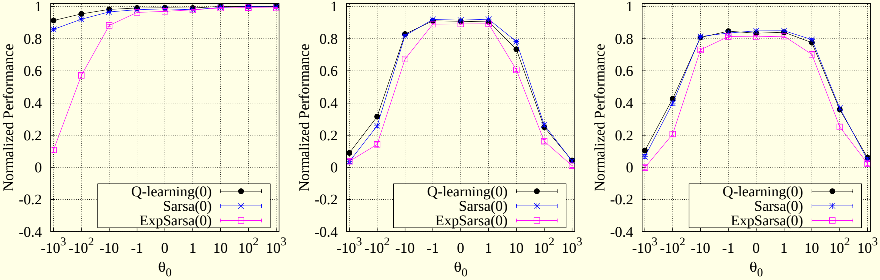
Parameter	**Controls:**	**Range**
$\#$*trials*	Samples per fitness evaluation	$[25, 250]$
$\#$*gens*	Generations	$[5, 50]$

Method-specific Parameter Search



(a) Stage 1 (b) Stage 2 (c) Stage 3 (d) Final solution

Effect of θ_0 on VF



(e) $\chi = 1, w = 1.$ (f) $\chi = 1, w = 5.$ (g) $\chi = 0.6, w = 5.$

Figure: [$s = 10, p = 0.2, \sigma = 0.$] Plots showing the effect of the initial weights θ_0 on the performance of on-line value function-based methods. Note the irregular spacing of points on the x axis. Plot (a) corresponds to an exact tabular representation with no generalization. Generalization is introduced in (b) by increasing w; additionally the expressiveness χ is reduced in (c).

Three Test Problem Instances

Table: Parameter settings for illustrative problem instances I_1, I_2, and I_3.

Problem instance	**s**	**p**	χ	**w**	σ
I_1	10	0.2	1	1	0
I_2	10	0.2	0.5	7	0
I_3	10	0.2	1	1	4

VF Representative

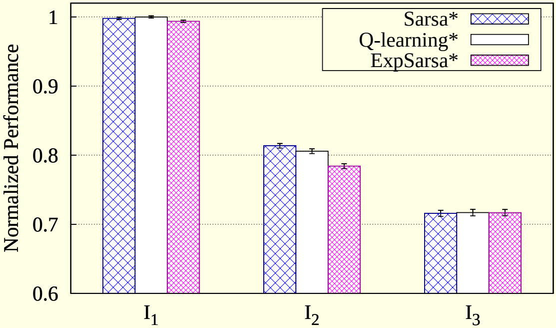


Figure: Comparison of the performance of different VF methods on the three problem instances from Table 4. Under each instance, and for each of the methods—Sarsa, Q-learning, and ExpSarsa—a systematic search identifies the method-specific parameter settings (α_0, ϵ_0, θ_0, and λ) yielding the highest performance after 50,000 episodes of training. The methods are marked "$*$" as they are run under these optimized parameter settings.

PS Representative

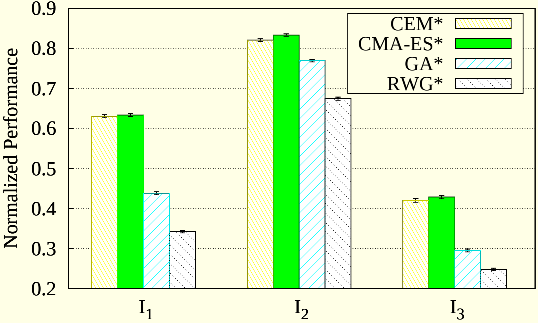
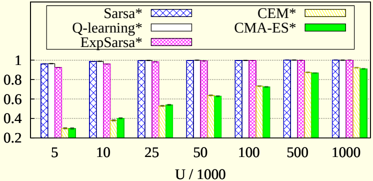
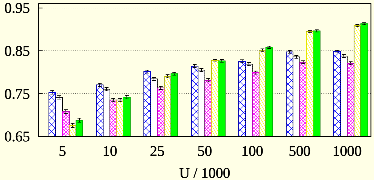


Figure: Comparison of the performance of different PS methods on the three problem instances from Table 4. Methods are marked "$*$" to denote that method-specific parameters—$\#$*trials* and $\#$*gens* (except $\#$*gens* for RWG)—have been optimized for each task instance.

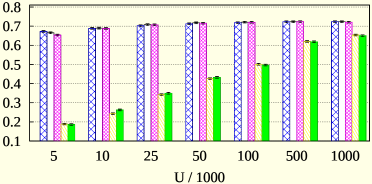
Setting U: Number of Training Episodes



(a) Normalized performance: $\mathbf{I_1}$



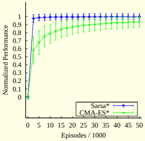
(b) Normalized performance: $\mathbf{I_2}$



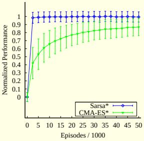
(c) Normalized performance: $\mathbf{I_3}$

Figure: Performance of different learning methods as the number of training episodes U is varied. Each plot corresponds to a problem instance from Table 4. Note the irregular spacing of points on the x axis. At each point, the best performance achieved by three learning methods from VF (Sarsa$*$, Q-learning$*$, and ExpSarsa$*$) and two from PS (CEM$*$, CMA-ES$*$) is shown (key specified in plot (a)).

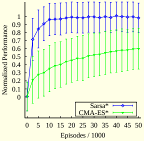
Effect of Problem Size and Stochasticity



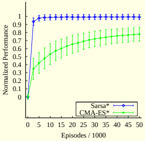
$s = \mathbf{6}, p = \mathbf{0}$



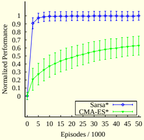
$s = \mathbf{6}, p = \mathbf{0.2}$



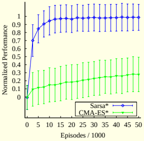
$s = \mathbf{6}, p = \mathbf{0.4}$



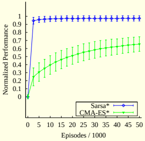
$s = \mathbf{10}, p = \mathbf{0}$



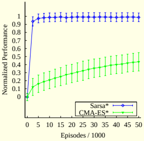
$s = \mathbf{10}, p = \mathbf{0.2}$



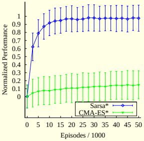
$s = \mathbf{10}, p = \mathbf{0.4}$



$s = \mathbf{14}, p = \mathbf{0}$

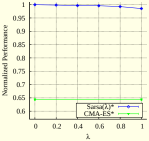


$s = \mathbf{14}, p = \mathbf{0.2}$

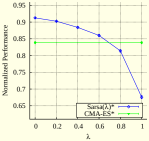


$s = \mathbf{14}, p = \mathbf{0.4}$

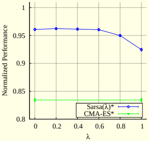
Effect of State Aliasing



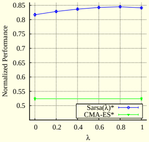
$\sigma = \mathbf{0}, w = \mathbf{1}$



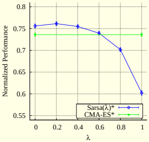
$\sigma = \mathbf{0}, w = \mathbf{5}$



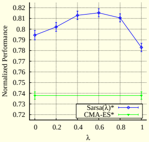
$\sigma = \mathbf{0}, w = \mathbf{9}$



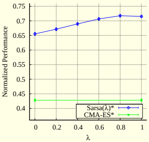
$\sigma = \mathbf{2}, w = \mathbf{1}$



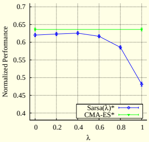
$\sigma = \mathbf{2}, w = \mathbf{5}$



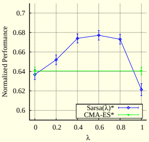
$\sigma = \mathbf{2}, w = \mathbf{9}$



$\sigma = \mathbf{4}, w = \mathbf{1}$

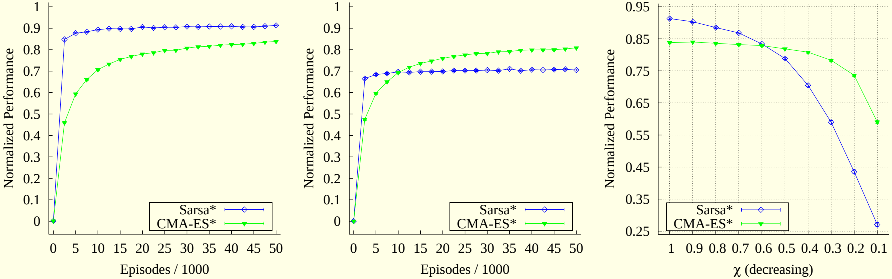


$\sigma = \mathbf{4}, w = \mathbf{5}$



$\sigma = \mathbf{4}, w = \mathbf{9}$

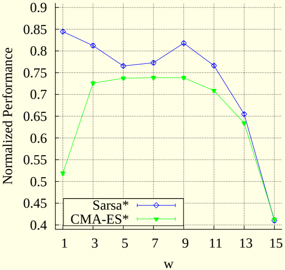
Effect of Expressiveness of Generalization Scheme



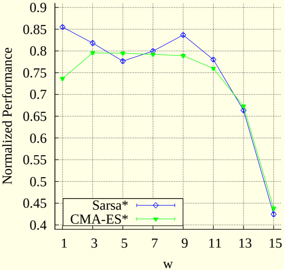
(a) $\chi = 1$ (b) $\chi = 0.4$ (c) Range of χ

Figure: [$s = 10$, $p = 0.2$, $w = 5$, $\sigma = 0$.] Plots (a) and (b) show learning curves of Sarsa(λ)* and CMA-ES* at different values of χ. Plot (c) shows the performance achieved after 50,000 episodes of training at different values of χ.

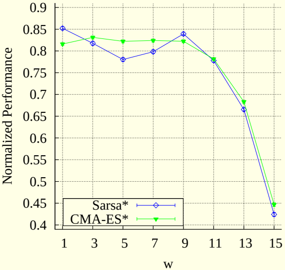
Effect of Generalization Width



(a) U = 50,000



(b) U = 500,000

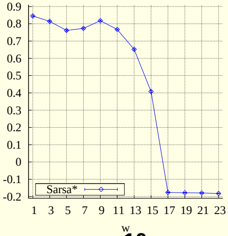


(c) U = 5,000,000

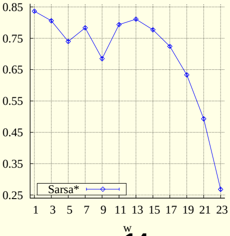
Figure: [$s = 10$, $p = 0.2$, $\chi = 1$, $\sigma = 2$.] Performance of Sarsa* and CMA-ES* at different values of w, optimized for (a) 50,000, (b) 500,000, and (c) 5,000,000 training episodes.

Effect of Generalization Width: Pattern

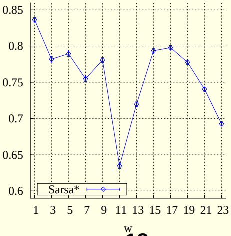
Normalized performance



$s = \mathbf{10}$

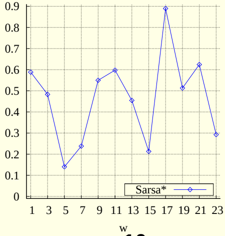


$s = \mathbf{14}$

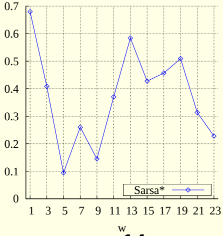


$s = \mathbf{18}$

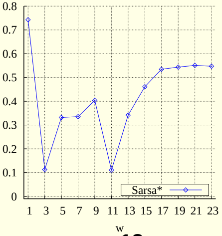
$\boldsymbol{\lambda}^*$



$s = \mathbf{10}$



$s = \mathbf{14}$



$s = \mathbf{18}$

Summary and Discussion

1. Parameterized learning problems: ideal methodology for future research.
2. Different horses (learning methods) for different courses (representations).
3. Generalization and optimistic initialization.
4. Automatic parameter tuning.
5. Integrate learning and representation discovery.

Retreat Presentation

1. provide a big picture motivation of the area
2. give a clear statement of the main objective
3. give an overview of important related work in the field
4. describe the key technical challenges
5. comment on the potential impact of solving them.

Thank You!