

Reinforcement Learning Toolbox Tutorial
Shortest Path Problem in a Grid-World
Institute for Theoretical Computer Science
TU-GRAZ

Gerhard Neumann

November 3, 2006

1 The Shortest Path Problem

In this quite simple learning problem we have to get from one or more given start states to one or more given target states in a grid-world. The source code for this tutorial can be found in the example directory of the toolbox.

This tutorial covers the use of grid-worlds in the toolbox. Grid-worlds are already implemented, so we don't have to define our own environment model. We will learn the shortest path problem for a given grid-world with a Q-Learning and a V-Learning algorithm.

2 Using Grid-worlds

The grid-worlds are modeled by the class *CGridworldModel*. This class loads a grid-world from a grid-world file. So how does a grid-world file look like? Here is the grid-world file for our example.

```
Gridworld Size: 10x10
StartValues: S
TargetValues: T
ProhibitedValues: X
```

```
XXXXXXXXXX
X.....X
X.TX....X
XXXX....X
X..X.XX..X
X.....X..X
XX.XXXXXXX
X.....X
X.....SX
XXXXXXXXXX
```

All grid-world files must have the same format. With the *StartValues*, *TargetValues* and the *ProhibitedValues* (walls) field we can set the symbols we want to use for these kinds of grid-world fields. The grid-world itself has to be in the specified size. All other symbols that are not listed in these 3 categories are considered to be "standard" grid-world fields. All fields with a start-value symbol are added to the list of possible start states. At the beginning of each episode, one start state is selected from this list randomly. Prohibited values represent the walls in the gridworld, whenever the agent runs in a wall, it stays at the same position and a negative reward is given. Whenever the agent reaches an target field, a positive reward is given and the episode is terminated. For the grid-world constructor we can additionally define the maximum number of bounces (the agent walks into a wall) before the episode gets reseted, we set this value to 50. The shortest path in this grid-world has 18 steps.

We also have the possibility to define individual rewards without writing an own reward function. We can set the standard reward (-0.2), the reward/punishment for bouncing into a wall (-5.0) and for reaching a target-state (100).

Additionally we have the possibility to set individual rewards for different symbols. This is only possible if the agent can reach a field with that symbol, so we can't set individual rewards for prohibited symbols, for all wall symbols the standard bounce reward is used. For all other symbols we can set the rewards with the function *setRewardForSymbol*, but we don't need that in our example.

The agent can not leave the grid-world, so even if we do not define walls at the borders of the grid-world, the agent will bounce against a "virtual" wall if he tries to leave the grid-world (so he will also get the negative reward for bouncing).

2.1 Using transition functions

The grid-world class implements the *CTransitionFunction* interface. A transition function implements the function $x_{t+1} = f(x_t, a_t)$. This transition function can be used to create the environment model, but it can additionally be used by learning algorithms which need the model of the environment (like V-Learning), which can not be done by a standard environment model class (*CEnvironmentModel*).

Having the transition function, we still need to define an environment model for the agent. The environment model consists of the current internal state of the agent, so the a transition function is not enough. But we can define our environment model easily with the class *CTransitionFunctionEnvironment* which takes a transition function as argument and creates an internal state for the agent.

```
// Create the Gridworld, set max_bounces to 50. The model will
// reset the episode if "max_bounces"
// bounces (walk into the wall) are reached.
CGridWorldModel *gridworldModel = new CGridWorldModel(
    gridworldFileName, 50);

// Set the reward values
// -0.2 for each move (in order to learn the shortest path)
// -5.0 for each bounce
// 100 for reaching the target state(s)

gridworldModel->setRewardStandard(-0.2);
```

```

gridworldModel->setRewardBounce(-5.0);
gridworldModel->setRewardSuccess(100.0);

// Create the environment for the agent, the environment saves the
// current state of the agent.
CEnvironmentModel *environmentModel = new
    CTransitionFunctionEnvironment(gridworldModel);

// the gridworld model implements the reward function too, so we
// can use this
CRewardFunction *rewardFunction = gridworldModel;

// Create the agent in our environmentModel.
CAgent *agent = new CAgent(environmentModel);

```

2.2 Creating the grid-world actions

For defining actions to move the agent in the grid-world, there is already the individual class *CGridWorldAction*. We have to specify the movement in the x and y direction for each action. It is recommended to use actions which have only a movement of 1 in either the x or the y direction. We create 4 actions for left, right, up and down. All these actions have to be added to the agent's action set.

```

// Add all possible Actions to the agent
// left
agent->addAction(new CGridWorldAction(-1,0));
// right
agent->addAction(new CGridWorldAction(1,0));
// up
agent->addAction(new CGridWorldAction(0,-1));
// down
agent->addAction(new CGridWorldAction(0,1));

```

2.3 The grid-world model state

The model state of the grid-world consists of 3 discrete state variables, the x- and the y-coordinates and the number of bounces. How do we use this model state for learning? There are several possibilities to do that, depending on what we need. For our case we need a global grid-world state, combining each $\langle x, y \rangle$ pair to a unique state number. This is easily done with the discretizer class *CGlobalGridworldDiscreteState*. There is an additional state (state number 0) which indicates the failed state (number of bounces has exceed its maximum), so the state size of our global grid-world state is always $size_x * size_y + 1$. This class only takes the size of the grid-world as parameters. As usual the discretizer has to be added to the agent's state modifier list.

```

// For the shortest path problem, we need a global state, i.e. each
// possible position in the grid is an own state

```

```

CAbstractStateDiscretizer *globalGridworldstate = new
    CGlobalGridWorldDiscreteState(gridworldModel->getSizeX() ,
        gridworldModel->getSizeY());

// In order to use the discretizer we have to add it to the agent's
    state modifier list.
// Always add your state modifiers to that list !!
agent->addStateModifier(globalGridworldstate);

```

But there are cases where we only want to use a local view of the current state. In the toolbox we have the possibility to use a 4, 4X (diagonal values) and a 8-neighborhood view. The corresponding classes are *CLocal4GridWorldState*, *CLocal4XGridWorldState*, and *CLocal8GridWorldState*. All this states consists of 4 resp. 8 discrete state variables, each of them having the discrete state size of the number of different symbols used in our grid-world file. We can merge this discrete state variables with the class *CModelStateDiscretizer* to one unique state index. If we don't want to use all possible values of the grid-world because of the large state size we can use the class *CLocalGridWorldDiscreteState*, here we can define which symbols we want to use from the grid-world, so for example we can define here that we only want to see wether there is a wall or not next to us. For further details see the class reference.

3 Using the Q-Value Learning Algorithm

The definition of our model is already finished, we can begin with the instantiation of our learning algorithm. We create a feature Q-Function with the global discrete state and the action set as parameters. Feature Q-Functions are used for discrete states and feature states (for linear approximators). Then the Q-Value Learner is created, we use a Sarsa Learner in this case (*CSarsaLearner*). The Sarsa learner needs the agent as additional argument. The agent object is used as estimation policy, so the learning algorithm always knows what the agent will do in the next step. As usual, we add the learner to the agent's listener list.

Now we still need an agent controller. We create a stochastic policy, which depends on our Q-Function (*CQStochasticPolicy*). As distribution we choose a ϵ -greedy policy with an ϵ value of 0.1 to guarantee exploration. For illustration we also specify an agent logger which logs the episodes and writes it to a file. This logger can then be used to load all episodes again from the file, and use these stored episodes for learning (see reference of the *CAgentLogger* class).

```

// Create an Agent Logger for logging the episodes
// Our agent logger logs the gridworld model state and the actions
    of the agent. This logger holds all episodes in memory.
CAgentLogger *logger = new CAgentLogger(gridworldModel->
    getStateProperties() , agent->getActions());
// add the logger to the agent's listener list
agent->addSemiMDPListener(logger);

// Q-Learning starts here

```

```

// Create our Q-Function, we will use a Feature Q-Function, which
// is table-like representation of the Q-Function.
// The Q-Function needs to know which actions and which state it
// has to use
CFeatureQFunction *qFunction = new CFeatureQFunction(agent->
    getActions(), globalGridworldstate);

// Create the Q-Function learner, we will use a SarsaLearner
// The Sarsa Learner needs the reward function, the Q-Function and
// the agent.
// The agent is used to get the estimation policy, because Sarsa
// Learning is On-Policy learning.
CSarsaLearner *qFunctionLearner = new CSarsaLearner(rewardFunction,
    qFunction, agent);

// Create the Controller for the agent from the QFunction. We will
// use a EpsilonGreedy-Policy for exploration.
CAgentController *qLearnerPolicy = new CQStochasticPolicy(agent->
    getActions(), new CEpsilonGreedyDistribution(0.1), qFunction);

// Set some options of the Etraces which are not default
vFunctionLearner->setParameter("ReplacingETraces", 1.0);
vFunctionLearner->setParameter("Lambda", 0.95);

// Add the learner to the agent listener list, so he can learn from
// the agent's steps.
agent->addSemiMDPLListener(qFunctionLearner);

// Set the controller of the agent
agent->setController(qLearnerPolicy);

```

Now we can start with the learning process, we learn 50 episodes with maximum 1000 steps per episode. The agent starts a new episodes if it has bounced into a wall for over 50 times.

3.1 Results of Q-Learning

Here is the listening of 3 trials of our program:

```

--< Reinforcement Learning Example - Learning the Shortest Path
Problem in a Grid-world >--

```

```

Episode 0 failed with 292 steps Episode 1 failed with 1000 steps
Episode 2 failed with 458 steps Episode 3 failed with 1000 steps
Episode 4 failed with 1000 steps Episode 5 failed with 1000 steps
Episode 6 failed with 1000 steps Episode 7 failed with 1000 steps
Episode 8 failed with 1000 steps Episode 9 failed with 1000 steps

```

Episode 10 failed with 1000 steps Episode 11 failed with 1000 steps
Episode 12 failed with 1000 steps Episode 13 failed with 1000 steps
Episode 14 failed with 1000 steps Episode 15 failed with 1000 steps
Episode 16 failed with 1000 steps Episode 17 failed with 1000 steps
Episode 18 failed with 1000 steps Episode 19 failed with 1000 steps
Episode 20 failed with 1000 steps Episode 21 failed with 1000 steps
Episode 22 failed with 1000 steps Episode 23 failed with 1000 steps
Episode 24 failed with 1000 steps Episode 25 failed with 1000 steps
Episode 26 failed with 1000 steps Episode 27 failed with 1000 steps
Episode 28 failed with 1000 steps Episode 29 failed with 1000 steps
Episode 30 failed with 1000 steps Episode 31 failed with 1000 steps
Episode 32 succeeded with 938 steps, 1 Episodes succeeded in the row
Episode 33 failed with 1000 steps Episode 34 failed with 1000 steps
Episode 35 succeeded with 635 steps, 1 Episodes succeeded in the row
Episode 36 failed with 1000 steps
Episode 37 succeeded with 177 steps, 1 Episodes succeeded in the row
Episode 38 succeeded with 170 steps, 2 Episodes succeeded in the row
Episode 39 succeeded with 47 steps, 3 Episodes succeeded in the row
Episode 40 succeeded with 60 steps, 4 Episodes succeeded in the row
Episode 41 succeeded with 25 steps, 5 Episodes succeeded in the row
Episode 42 succeeded with 20 steps, 6 Episodes succeeded in the row
Episode 43 succeeded with 18 steps, 7 Episodes succeeded in the row
Episode 44 succeeded with 19 steps, 8 Episodes succeeded in the row
Episode 45 succeeded with 18 steps, 9 Episodes succeeded in the row
Episode 46 succeeded with 18 steps, 10 Episodes succeeded in the row
Episode 47 succeeded with 18 steps, 11 Episodes succeeded in the row
Episode 48 succeeded with 18 steps, 12 Episodes succeeded in the row
Episode 49 succeeded with 23 steps, 13 Episodes succeeded in the row

--< Reinforcement Learning Example - Learning the Shortest Path
Problem in a Grid-world >--

Episode 0 failed with 357 steps Episode 1 failed with 1000 steps
Episode 2 failed with 1000 steps Episode 3 failed with 1000 steps
Episode 4 failed with 620 steps Episode 5 failed with 1000 steps
Episode 6 failed with 1000 steps Episode 7 failed with 1000 steps
Episode 8 failed with 1000 steps Episode 9 failed with 1000 steps
Episode 10 failed with 1000 steps Episode 11 failed with 1000 steps
Episode 12 failed with 1000 steps Episode 13 failed with 1000 steps
Episode 14 failed with 1000 steps Episode 15 failed with 1000 steps
Episode 16 failed with 1000 steps Episode 17 failed with 1000 steps
Episode 18 failed with 1000 steps Episode 19 failed with 1000 steps
Episode 20 failed with 1000 steps Episode 21 failed with 1000 steps
Episode 22 failed with 1000 steps Episode 23 failed with 1000 steps
Episode 24 failed with 1000 steps Episode 25 failed with 1000 steps
Episode 26 failed with 1000 steps

Episode 27 succeeded with 341 steps, 1 Episodes succeeded in the row
Episode 28 succeeded with 192 steps, 2 Episodes succeeded in the row
Episode 29 succeeded with 170 steps, 3 Episodes succeeded in the row
Episode 30 succeeded with 52 steps, 4 Episodes succeeded in the row
Episode 31 succeeded with 95 steps, 5 Episodes succeeded in the row
Episode 32 succeeded with 29 steps, 6 Episodes succeeded in the row
Episode 33 succeeded with 25 steps, 7 Episodes succeeded in the row
Episode 34 succeeded with 29 steps, 8 Episodes succeeded in the row
Episode 35 succeeded with 23 steps, 9 Episodes succeeded in the row
Episode 36 succeeded with 33 steps, 10 Episodes succeeded in the row
Episode 37 succeeded with 20 steps, 11 Episodes succeeded in the row
Episode 38 succeeded with 20 steps, 12 Episodes succeeded in the row
Episode 39 succeeded with 23 steps, 13 Episodes succeeded in the row
Episode 40 succeeded with 41 steps, 14 Episodes succeeded in the row
Episode 41 succeeded with 21 steps, 15 Episodes succeeded in the row
Episode 42 succeeded with 30 steps, 16 Episodes succeeded in the row
Episode 43 succeeded with 26 steps, 17 Episodes succeeded in the row
Episode 44 succeeded with 24 steps, 18 Episodes succeeded in the row
Episode 45 succeeded with 20 steps, 19 Episodes succeeded in the row
Episode 46 succeeded with 20 steps, 20 Episodes succeeded in the row
Episode 47 succeeded with 25 steps, 21 Episodes succeeded in the row
Episode 48 succeeded with 24 steps, 22 Episodes succeeded in the row
Episode 49 succeeded with 27 steps, 23 Episodes succeeded in the row

--< Reinforcement Learning Example - Learning the Shortest Path
Problem in a Grid-world >--

Episode 0 failed with 150 steps Episode 1 failed with 589 steps
Episode 2 failed with 1000 steps Episode 3 failed with 1000 steps
Episode 4 failed with 1000 steps Episode 5 failed with 1000 steps
Episode 6 failed with 1000 steps Episode 7 failed with 1000 steps
Episode 8 succeeded with 459 steps, 1 Episodes succeeded in the row
Episode 9 failed with 1000 steps Episode 10 failed with 1000 steps
Episode 11 succeeded with 494 steps, 1 Episodes succeeded in the row
Episode 12 succeeded with 47 steps, 2 Episodes succeeded in the row
Episode 13 succeeded with 139 steps, 3 Episodes succeeded in the row
Episode 14 succeeded with 59 steps, 4 Episodes succeeded in the row
Episode 15 succeeded with 31 steps, 5 Episodes succeeded in the row
Episode 16 succeeded with 26 steps, 6 Episodes succeeded in the row
Episode 17 succeeded with 27 steps, 7 Episodes succeeded in the row
Episode 18 succeeded with 26 steps, 8 Episodes succeeded in the row
Episode 19 succeeded with 18 steps, 9 Episodes succeeded in the row
Episode 20 succeeded with 20 steps, 10 Episodes succeeded in the row
Episode 21 succeeded with 30 steps, 11 Episodes succeeded in the row
Episode 22 succeeded with 21 steps, 12 Episodes succeeded in the row
Episode 23 succeeded with 18 steps, 13 Episodes succeeded in the row

Episode 24 succeeded with 19 steps, 14 Episodes succeeded in the row
Episode 25 succeeded with 26 steps, 15 Episodes succeeded in the row
Episode 26 succeeded with 24 steps, 16 Episodes succeeded in the row
Episode 27 succeeded with 20 steps, 17 Episodes succeeded in the row
Episode 28 succeeded with 24 steps, 18 Episodes succeeded in the row
Episode 29 succeeded with 18 steps, 19 Episodes succeeded in the row
Episode 30 succeeded with 20 steps, 20 Episodes succeeded in the row
Episode 31 succeeded with 18 steps, 21 Episodes succeeded in the row
Episode 32 succeeded with 19 steps, 22 Episodes succeeded in the row
Episode 33 succeeded with 18 steps, 23 Episodes succeeded in the row
Episode 34 succeeded with 23 steps, 24 Episodes succeeded in the row
Episode 35 succeeded with 21 steps, 25 Episodes succeeded in the row
Episode 36 succeeded with 18 steps, 26 Episodes succeeded in the row
Episode 37 succeeded with 25 steps, 27 Episodes succeeded in the row
Episode 38 succeeded with 26 steps, 28 Episodes succeeded in the row
Episode 39 succeeded with 18 steps, 29 Episodes succeeded in the row
Episode 40 succeeded with 28 steps, 30 Episodes succeeded in the row
Episode 41 succeeded with 18 steps, 31 Episodes succeeded in the row
Episode 42 succeeded with 24 steps, 32 Episodes succeeded in the row
Episode 43 succeeded with 23 steps, 33 Episodes succeeded in the row
Episode 44 succeeded with 19 steps, 34 Episodes succeeded in the row
Episode 45 succeeded with 22 steps, 35 Episodes succeeded in the row
Episode 46 succeeded with 26 steps, 36 Episodes succeeded in the row
Episode 47 succeeded with 24 steps, 37 Episodes succeeded in the row
Episode 48 succeeded with 18 steps, 38 Episodes succeeded in the row
Episode 49 succeeded with 24 steps, 39 Episodes succeeded in the row

How fast the agent finds the way to the target depends on the exploration, so the learning speed is quite different in the learning trials. At the beginning some episodes fail before reaching 1000 steps because the agent has bounced into a wall more than 50 times. The optimal path of 18 steps could only be reached in few episodes because of the exploration value of 10%. Adjusting the epsilon value during the learning trial would solve that problem. See *CAaptiveParameterCalculator* how this is done easily.

4 Using the V-Learning Algorithm

V-Learner just learn the V-Values instead of the Q-Values. The policy is then created with a 1 step state prediction for each action, calculating the V-Value of the predicted successor state. Since we use a transition function interface, this state prediction can be easily done with the grid-world model.

In order to use this algorithm we have again only to replace the algorithm part. Now we create a feature V-Function (*CFeatureVFunction*) instead of the Q-Function and a V-Learner (*CVFunctionLearner*) instead of a Sarsa Learner. V-Function learning algorithms also do not need any estimation policy, they always estimate the value of the current policy (on-policy learning).

Now we cannot use a policy which depends on Q-Values. We need a policy which does the state prediction as mentioned above. This policy is represented by the class *CVMStochas-*

ticPolicy, which is again a stochastic policy (so it can be used the same way as the Q-Policy), but now the goodness of an action is calculated via the state prediction. The policy takes the V-Function, the grid-world model as transition function, the reward function (which is also our grid-world model object) and a list off all state modifiers which are used by the V-Function (it is recommended to use the agent's state modifier list for simplicity), as additional parameters. As action distribution we choose again an epsilon greedy distribution with $\epsilon = 0.1$.

Now we can start the learning process again, we have the same procedure like in the Q-Learning program.

```
// Create our V-Function, we will use a Feature V-Function, which
// is table-like representation of the V-Function.
// The Q-Function needs to know which state it has to use
CFeatureVFunction *vFunction = new CFeatureVFunction(
    globalGridworldstate);

// Create the V-Function learner, we will use a standard TD-Learner
// The V-Function Learner needs the reward function and of course
// the V-Function.
CVFunctionLearner *vFunctionLearner = new CVFunctionLearner(
    rewardFunction, vFunction);

// Create the Controller for the agent from the VFunction and the
// gridworld model as state predictor.
// Additionally to the V-Function the policy needs the gridworld
// model to calculate the next states for all actions
// and the reward function to calculate the rewards for that states
// . It also needs the state modifiers list from the agent to get
// the value from the V-Function.
// We will use a EpsilonGreedy-Policy for exploration.
CAgentController *vLearnerPolicy = new CVMStochasticPolicy(agent->
    getActions(), new CEpsilonGreedyDistribution(0.1), vFunction,
    gridworldModel, rewardFunction, agent->getStateModifiers());

// Set some options of the Etraces which are not default
vFunctionLearner->setParameter("ReplacingETraces", 1.0);
vFunctionLearner->setParameter("Lambda", 0.95);

// Add the learner to the agent listener list, so he can learn from
// the agent's steps.
agent->addSemiMDPListener(vFunctionLearner);

// Set the controller of the agent
agent->setController(vLearnerPolicy);
```

4.1 Results of V-Learning

Here is the listing of 3 trials of V-Learning:

```
--< Reinforcement Learning Example - Learning the Shortest Path  
Problem in a Grid-world >=
```

```
Episode 0 succeeded with 48 steps, 1 Episodes succeeded in the row  
Episode 1 succeeded with 29 steps, 2 Episodes succeeded in the row  
Episode 2 succeeded with 32 steps, 3 Episodes succeeded in the row  
Episode 3 succeeded with 36 steps, 4 Episodes succeeded in the row  
Episode 4 succeeded with 32 steps, 5 Episodes succeeded in the row  
Episode 5 succeeded with 30 steps, 6 Episodes succeeded in the row  
Episode 6 succeeded with 27 steps, 7 Episodes succeeded in the row  
Episode 7 succeeded with 21 steps, 8 Episodes succeeded in the row  
Episode 8 succeeded with 25 steps, 9 Episodes succeeded in the row  
Episode 9 succeeded with 18 steps, 10 Episodes succeeded in the row  
Episode 10 succeeded with 25 steps, 11 Episodes succeeded in the row  
Episode 11 succeeded with 23 steps, 12 Episodes succeeded in the row  
Episode 12 succeeded with 22 steps, 13 Episodes succeeded in the row  
Episode 13 succeeded with 23 steps, 14 Episodes succeeded in the row  
Episode 14 succeeded with 20 steps, 15 Episodes succeeded in the row  
Episode 15 succeeded with 20 steps, 16 Episodes succeeded in the row  
Episode 16 succeeded with 22 steps, 17 Episodes succeeded in the row  
Episode 17 succeeded with 22 steps, 18 Episodes succeeded in the row  
Episode 18 succeeded with 23 steps, 19 Episodes succeeded in the row  
Episode 19 succeeded with 23 steps, 20 Episodes succeeded in the row  
Episode 20 succeeded with 22 steps, 21 Episodes succeeded in the row  
Episode 21 succeeded with 20 steps, 22 Episodes succeeded in the row  
Episode 22 succeeded with 21 steps, 23 Episodes succeeded in the row  
Episode 23 succeeded with 23 steps, 24 Episodes succeeded in the row  
Episode 24 succeeded with 24 steps, 25 Episodes succeeded in the row  
Episode 25 succeeded with 20 steps, 26 Episodes succeeded in the row  
Episode 26 succeeded with 25 steps, 27 Episodes succeeded in the row  
Episode 27 succeeded with 22 steps, 28 Episodes succeeded in the row  
Episode 28 succeeded with 23 steps, 29 Episodes succeeded in the row  
Episode 29 succeeded with 21 steps, 30 Episodes succeeded in the row  
Episode 30 succeeded with 21 steps, 31 Episodes succeeded in the row  
Episode 31 succeeded with 20 steps, 32 Episodes succeeded in the row  
Episode 32 succeeded with 24 steps, 33 Episodes succeeded in the row  
Episode 33 succeeded with 21 steps, 34 Episodes succeeded in the row  
Episode 34 succeeded with 27 steps, 35 Episodes succeeded in the row  
Episode 35 succeeded with 20 steps, 36 Episodes succeeded in the row  
Episode 36 succeeded with 22 steps, 37 Episodes succeeded in the row  
Episode 37 succeeded with 20 steps, 38 Episodes succeeded in the row  
Episode 38 succeeded with 20 steps, 39 Episodes succeeded in the row  
Episode 39 succeeded with 20 steps, 40 Episodes succeeded in the row
```

Episode 40 succeeded with 21 steps, 41 Episodes succeeded in the row
Episode 41 succeeded with 18 steps, 42 Episodes succeeded in the row
Episode 42 succeeded with 24 steps, 43 Episodes succeeded in the row
Episode 43 succeeded with 25 steps, 44 Episodes succeeded in the row
Episode 44 succeeded with 20 steps, 45 Episodes succeeded in the row
Episode 45 succeeded with 22 steps, 46 Episodes succeeded in the row
Episode 46 succeeded with 27 steps, 47 Episodes succeeded in the row
Episode 47 succeeded with 22 steps, 48 Episodes succeeded in the row
Episode 48 succeeded with 22 steps, 49 Episodes succeeded in the row
Episode 49 succeeded with 26 steps, 50 Episodes succeeded in the row

--< Reinforcement Learning Example - Learning the Shortest Path
Problem in a Grid-world >=--

Episode 0 succeeded with 111 steps, 1 Episodes succeeded in the row
Episode 1 succeeded with 39 steps, 2 Episodes succeeded in the row
Episode 2 succeeded with 38 steps, 3 Episodes succeeded in the row
Episode 3 succeeded with 26 steps, 4 Episodes succeeded in the row
Episode 4 succeeded with 26 steps, 5 Episodes succeeded in the row
Episode 5 succeeded with 18 steps, 6 Episodes succeeded in the row
Episode 6 succeeded with 20 steps, 7 Episodes succeeded in the row
Episode 7 succeeded with 20 steps, 8 Episodes succeeded in the row
Episode 8 succeeded with 24 steps, 9 Episodes succeeded in the row
Episode 9 succeeded with 20 steps, 10 Episodes succeeded in the row
Episode 10 succeeded with 19 steps, 11 Episodes succeeded in the row
Episode 11 succeeded with 21 steps, 12 Episodes succeeded in the row
Episode 12 succeeded with 22 steps, 13 Episodes succeeded in the row
Episode 13 succeeded with 22 steps, 14 Episodes succeeded in the row
Episode 14 succeeded with 21 steps, 15 Episodes succeeded in the row
Episode 15 succeeded with 18 steps, 16 Episodes succeeded in the row
Episode 16 succeeded with 20 steps, 17 Episodes succeeded in the row
Episode 17 succeeded with 18 steps, 18 Episodes succeeded in the row
Episode 18 succeeded with 19 steps, 19 Episodes succeeded in the row
Episode 19 succeeded with 18 steps, 20 Episodes succeeded in the row
Episode 20 succeeded with 23 steps, 21 Episodes succeeded in the row
Episode 21 succeeded with 18 steps, 22 Episodes succeeded in the row
Episode 22 succeeded with 20 steps, 23 Episodes succeeded in the row
Episode 23 succeeded with 19 steps, 24 Episodes succeeded in the row
Episode 24 succeeded with 21 steps, 25 Episodes succeeded in the row
Episode 25 succeeded with 20 steps, 26 Episodes succeeded in the row
Episode 26 succeeded with 22 steps, 27 Episodes succeeded in the row
Episode 27 succeeded with 18 steps, 28 Episodes succeeded in the row
Episode 28 succeeded with 18 steps, 29 Episodes succeeded in the row
Episode 29 succeeded with 20 steps, 30 Episodes succeeded in the row
Episode 30 succeeded with 22 steps, 31 Episodes succeeded in the row
Episode 31 succeeded with 24 steps, 32 Episodes succeeded in the row
Episode 32 succeeded with 19 steps, 33 Episodes succeeded in the row

Episode 33 succeeded with 20 steps, 34 Episodes succeeded in the row
Episode 34 succeeded with 20 steps, 35 Episodes succeeded in the row
Episode 35 succeeded with 21 steps, 36 Episodes succeeded in the row
Episode 36 succeeded with 21 steps, 37 Episodes succeeded in the row
Episode 37 succeeded with 18 steps, 38 Episodes succeeded in the row
Episode 38 succeeded with 24 steps, 39 Episodes succeeded in the row
Episode 39 succeeded with 19 steps, 40 Episodes succeeded in the row
Episode 40 succeeded with 21 steps, 41 Episodes succeeded in the row
Episode 41 succeeded with 20 steps, 42 Episodes succeeded in the row
Episode 42 succeeded with 20 steps, 43 Episodes succeeded in the row
Episode 43 succeeded with 22 steps, 44 Episodes succeeded in the row
Episode 44 succeeded with 23 steps, 45 Episodes succeeded in the row
Episode 45 succeeded with 18 steps, 46 Episodes succeeded in the row
Episode 46 succeeded with 21 steps, 47 Episodes succeeded in the row
Episode 47 succeeded with 18 steps, 48 Episodes succeeded in the row
Episode 48 succeeded with 21 steps, 49 Episodes succeeded in the row
Episode 49 succeeded with 27 steps, 50 Episodes succeeded in the row

--< Reinforcement Learning Example - Learning the Shortest Path
Problem in a Grid-world >=

Episode 0 succeeded with 204 steps, 1 Episodes succeeded in the row
Episode 1 succeeded with 33 steps, 2 Episodes succeeded in the row
Episode 2 succeeded with 22 steps, 3 Episodes succeeded in the row
Episode 3 succeeded with 35 steps, 4 Episodes succeeded in the row
Episode 4 succeeded with 24 steps, 5 Episodes succeeded in the row
Episode 5 succeeded with 21 steps, 6 Episodes succeeded in the row
Episode 6 succeeded with 21 steps, 7 Episodes succeeded in the row
Episode 7 succeeded with 22 steps, 8 Episodes succeeded in the row
Episode 8 succeeded with 20 steps, 9 Episodes succeeded in the row
Episode 9 succeeded with 18 steps, 10 Episodes succeeded in the row
Episode 10 succeeded with 22 steps, 11 Episodes succeeded in the row
Episode 11 succeeded with 18 steps, 12 Episodes succeeded in the row
Episode 12 succeeded with 19 steps, 13 Episodes succeeded in the row
Episode 13 succeeded with 18 steps, 14 Episodes succeeded in the row
Episode 14 succeeded with 19 steps, 15 Episodes succeeded in the row
Episode 15 succeeded with 27 steps, 16 Episodes succeeded in the row
Episode 16 succeeded with 26 steps, 17 Episodes succeeded in the row
Episode 17 succeeded with 23 steps, 18 Episodes succeeded in the row
Episode 18 succeeded with 21 steps, 19 Episodes succeeded in the row
Episode 19 succeeded with 18 steps, 20 Episodes succeeded in the row
Episode 20 succeeded with 20 steps, 21 Episodes succeeded in the row
Episode 21 succeeded with 25 steps, 22 Episodes succeeded in the row
Episode 22 succeeded with 18 steps, 23 Episodes succeeded in the row
Episode 23 succeeded with 22 steps, 24 Episodes succeeded in the row
Episode 24 succeeded with 18 steps, 25 Episodes succeeded in the row

Episode 25 succeeded with 20 steps, 26 Episodes succeeded in the row
Episode 26 succeeded with 19 steps, 27 Episodes succeeded in the row
Episode 27 succeeded with 22 steps, 28 Episodes succeeded in the row
Episode 28 succeeded with 20 steps, 29 Episodes succeeded in the row
Episode 29 succeeded with 19 steps, 30 Episodes succeeded in the row
Episode 30 succeeded with 23 steps, 31 Episodes succeeded in the row
Episode 31 succeeded with 20 steps, 32 Episodes succeeded in the row
Episode 32 succeeded with 20 steps, 33 Episodes succeeded in the row
Episode 33 succeeded with 23 steps, 34 Episodes succeeded in the row
Episode 34 succeeded with 18 steps, 35 Episodes succeeded in the row
Episode 35 succeeded with 20 steps, 36 Episodes succeeded in the row
Episode 36 succeeded with 20 steps, 37 Episodes succeeded in the row
Episode 37 succeeded with 19 steps, 38 Episodes succeeded in the row
Episode 38 succeeded with 19 steps, 39 Episodes succeeded in the row
Episode 39 succeeded with 20 steps, 40 Episodes succeeded in the row
Episode 40 succeeded with 21 steps, 41 Episodes succeeded in the row
Episode 41 succeeded with 21 steps, 42 Episodes succeeded in the row
Episode 42 succeeded with 22 steps, 43 Episodes succeeded in the row
Episode 43 succeeded with 19 steps, 44 Episodes succeeded in the row
Episode 44 succeeded with 19 steps, 45 Episodes succeeded in the row
Episode 45 succeeded with 25 steps, 46 Episodes succeeded in the row
Episode 46 succeeded with 20 steps, 47 Episodes succeeded in the row
Episode 47 succeeded with 21 steps, 48 Episodes succeeded in the row
Episode 48 succeeded with 19 steps, 49 Episodes succeeded in the row
Episode 49 succeeded with 22 steps, 50 Episodes succeeded in the row

As we can see the results are much better than with Q-Learning, which is not really surprising.