1 On-policy Prediction with Approximation

1.1 Exercise 9.1

1.1.1 Q

Show that tabular methods such as presented in Part I of this book are a special case of linear function approximation. What would the feature vectors be?

1.1.2 A

Write $\hat{V}(s, \mathbf{w}) = w$ and we get that $\nabla_{\mathbf{w}} \hat{V}(s, \mathbf{w}) = 1$ so we return to tabular TD learning. In this case the features are $x(s) = 1 \; \forall s \in \mathcal{S}$.

1.2 Exercise 9.2

1.2.1 Q

Why does (9.17) define $(n+1)^k$ distinct features for dimension k?

1.2.2 A

Each of the k terms can independently have one of $n+1$ exponents, hence the total number of features is $(n+1)^k$.

1.3 Exercise 9.3

1.3.1 Q

What n and $c_{i,j}$ produce the feature vectors $\mathbf{x}(s) = (1, s_1, s_2, s_1 s_2, s_1^2, s_2^2, s_1 s_2^2, s_1^2 s_2, s_1^2 s_2^2)$?

1.3.2 A

$n = 2$ and $c_i, j = C_{ij}$ where

$$C = \begin{pmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \\ 1 & 1 \\ 2 & 0 \\ 0 & 2 \\ 1 & 2 \\ 2 & 1 \\ 2 & 2 \end{pmatrix}$$

1.4 Exercise 9.4

1.4.1 Q

Suppose we believe that one of two state dimensions is more likely to have an effect on the value function than is the other, that generalization should be primarily across this dimension rather than along it. What kind of tilings could be used to take advantage of this prior knowledge?

1.4.2 A

Tiles that are thin along the dimension of interest and long across it. Rectangles, for instance.

1.5 Exercise 9.5

1.5.1 Q

Suppose you are using tile coding to transform a seven-dimensional continuous state space into binary feature vectors to estimate a state value function $\hat{v}(s, \mathbf{w}) \approx v_\pi(s)$. You believe that the dimensions do not interact strongly, so you decide to use eight tilings of each dimension separately (stripe tilings), for $7 \times 8 = 56$ tilings. In addition, in case there are some pairwise interactions between the dimensions, you also take all $\binom{7}{2} = 21$ pairs of dimensions and tile each pair conjunctively with rectangular tiles. You make two tilings for each pair of dimensions, making a grand total of $21 \times 2 + 56 = 98$ tilings. Given these feature vectors, you suspect that you still have to average out some noise, so you decide that you want learning to be gradual, taking about 10 presentations with the same feature vector before learning nears its asymptote. What step-size parameter α should you use? Why?

1.5.2 A

Each tiling is a partition, so each tiling has exactly one tile activated per state. This means that in our case the number of features is 98. We consider each of these equally likely because we are uninformed. We therefore take

$$\alpha = \frac{1}{10 \times 98} = \frac{1}{980}.$$

So that on average we see each feature 10 times before asymptote. [Note that this assumes a constant target.]