Einführung in die
Künstliche Intelligenz
WS10/11 - Prof. Dr. J. Fürnkranz

TECHNISCHE UNIVERSITÄT DARMSTADT

Beispiellösung für das 7. Übungsblatt (16.02.2011)

Aufgabe 1 Reinforcement Learning

a) Laut Aufgabenstellung erhalten nur (unmittelbare) Aktionen einen Reward (1), die zur Folge haben, dass der Agent sich daraufhin im Feld f befindet. Diese Aktionen sind im Zustand c nach unten zu gehen und im Zustand e sich nach rechts zu bewegen. Alle anderen Aktionen erhalten einen Reward von 0.

$$\begin{array}{lllll} r(a,r)=0 & r(b,r)=0 & r(c,u)=1 & r(d,o)=0 & r(e,o)=0 \\ r(a,u)=0 & r(b,u)=0 & r(c,l)=0 & r(d,r)=0 & r(e,r)=1 \\ & r(b,l)=0 & & & r(e,l)=0 \end{array}$$

b) Der akkumulierte erwartete Reward eines Zustandes s wird folgendermaßen berechnet: $V^\pi(s) = \sum_{k=0}^{\infty} \gamma^k \cdot r_k$, wobei die Rewards r_k, den Rewards entsprechen, die man erhält, wenn man vom Anfangszustand s aus Aktionen gemäß der Policy π ausführt. Als Beispiel wird die Berechnung von $V^\pi(d)$ dargestellt: Laut Policy bewegt sich der Agent ausgehend von d wie folgt: $\rightarrow a \rightarrow b \rightarrow c \rightarrow f$, wobei im Feld f keine Aktion mehr möglich ist.

Die Bewertung $V^\pi(d)$ ergibt sich also als:

$$\begin{aligned} V^\pi(d) &= \gamma^0 \cdot r(d,o) + \gamma^1 \cdot r(a,r) + \gamma^2 \cdot r(b,r) + \gamma^3 \cdot r(c,u) \\ &= 1 \cdot 0 + 0.8 \cdot 0 + 0.8^2 \cdot 0 + 0.8^3 \cdot 1 \\ &= 0.512 \end{aligned}$$

Analog berechnet man die Bewertungen der restlichen Felder:

$V^\pi(a) = 0.64$	$V^\pi(b) = 0.8$	$V^\pi(c) = 1$
$V^\pi(d) = 0.512$	$V^\pi(e) = 0.64$	

c) PolicyImprovement ändert die aktuelle Policy π für einen Zustand s um, indem sie die Aktion a selektiert, die folgendes maximiert:

$$\max_a r(s,a) + \gamma \cdot V^\pi(s') \qquad \text{wobei } s' = \delta(s,a)$$

Die aktuelle Policy $\pi(e)$ für den Zustand e würde einen Schritt nach **oben** vorgeben, mit der Gesamt-Bewertung 0.64. Für die anderen Aktionen ergibt sich:

links: $r(e,l) + \gamma \cdot V^\pi(d) = 0 + 0.8 \cdot 0.512 = 0.4096$
rechts: $r(e,r) = 1$

Da die Aktion *rechts* im Zustand e die beste Bewertung hat, würde die aktuelle Policy für den Zustand e mit der Anweisung $\pi'(e) = r$ überschrieben werden.

d) Wir überlegen uns für jedes Feld, welches ein optimaler Weg zu f wäre. Beispielsweise würde für das Feld a der Weg $\rightarrow b \rightarrow c \rightarrow f$ einen optimalen Reward erzielen, genauso wie $\rightarrow d \rightarrow e \rightarrow f$, nämlich:

$$= 0.64$$

Analog berechnet man die Bewertungen der restlichen Felder und erhält:

$V^*(a) = 0.64$	$V^*(b) = 0.8$	$V^*(c) = 1$
$V^*(d) = 0.8$	$V^*(e) = 1$	

Die optimale Q-Funktion für alle Zustandspaare lässt sich nun mit den berechneten optimalen Bewertungsfunktionen recht einfach berechnen. Wie aus der Vorlesung bekannt, gilt für die optimale Q-Funktion :

$$Q(s,a) = r(s,a) + \gamma \cdot V^*(s')$$

Im Feld a erhalten wir beispielsweise :
$Q(a,r) = r(a,r) + \gamma \cdot V^*(b) = 0 + 0.8 \cdot 0.8 = 0.64$
$Q(a,u) = r(a,u) + \gamma \cdot V^*(d) = 0 + 0.8 \cdot 0.8 = 0.64$
Insgesamt ergibt dies:

$$\begin{array}{lllll} Q(a,r) = 0.64 & Q(b,r) = 0.8 & Q(c,u) = 1 & Q(d,o) = 0.512 & Q(e,o) = 0.64 \\ Q(a,u) = 0.64 & Q(b,u) = 0.8 & Q(c,l) = 0.64 & Q(d,r) = 0.8 & Q(e,r) = 1 \\ & Q(b,l) = 0.512 & & & Q(e,l) = 0.64 \end{array}$$

e) Die optimale Policy wählt in jedem Feld diejenige Aktion aus, die den höchsten akkumulierten erwarteten Reward verspricht:

$$\begin{aligned} \pi^*(s) &= \text{argmax}_a\, r(s,a) + \gamma \cdot V^*(s') \\ &= \text{argmax}_a\, Q(s,a) \end{aligned}$$

Mithilfe der vorigen Teilaufgaben lässt sich einfach die optimale Policy ablesen, indem man für jeden Zustand die Aktion wählt, die den höchsten Q-Wert aufweist. Insgesamt ergibt das folgende graphisch dargestellte Policy:

$\downarrow\rightarrow$	$\downarrow\rightarrow$	$\downarrow$
$\rightarrow$	$\rightarrow$	

f) Alle Werte $\hat{Q}(s,a)$ werden zunächst auf null gesetzt. Wir verwenden folgende graphische Ansicht der $\hat{Q}$-Werte:

a $\hat{Q}(a,r) = 0$ $\hat{Q}(a,u) = 0$	$\hat{Q}(b,l) = 0$ b $\hat{Q}(b,r) = 0$ $\hat{Q}(b,u) = 0$	$\hat{Q}(c,l) = 0$ c $\hat{Q}(c,u) = 0$
$\hat{Q}(d,o) = 0$ d $\hat{Q}(d,r) = 0$	$\hat{Q}(e,o) = 0$ $\hat{Q}(e,l) = 0$ e $\hat{Q}(e,r) = 0$	$\hat{Q}(f,o) = 0$ $\hat{Q}(f,l) = 0$ f

Wir wählen zufällig ein Feld aus, sagen wir d. Da die beiden Aktionen o und r gleich bewertet sind, wählen wir erneut zufällig die Aktion r.

Nun ergibt sich der neue Wert $\hat{Q}(d,r) = \hat{Q}(d,r) + \alpha[r(d,r) + \gamma \cdot \max_a \hat{Q}(e,a) - \hat{Q}(d,r)]$. Da α auf 1 gesetzt wurde, wird der alte Wert nicht berücksichtigt, d.h. als Update-Regel wird $\hat{Q}(d,r) = r(d,r) + \gamma \cdot \max_a \hat{Q}(e,a)$ verwendet.

Darüber hinaus sind alle $\hat{Q}$-Werte von e auf 0 gesetzt, so dass $\hat{Q}(d,r) = 0 + 0.8 \cdot 0 = 0$.

Im Feld e wählen wir zufällig die Aktion r : $\hat{Q}(e,r) = 1$

Die momentanen $\hat{Q}$-Werte sehen dann wie folgt aus:

a 0.0 0.0	0.0 b 0.0 0.0	0.0 c 0.0
0.0 d 0.0	0.0 0.0 e 1.0	0.0 0.0 f

In einer weiteren Iteration starten wir von b und wählen die Aktion u. Es ergibt sich nun $\hat{Q}(b,u) = r(b,u) + \gamma \cdot \max_a \hat{Q}(e,a)$. Laut unserer aktuellen Q-Funktion ist im Feld e die optimale Aktion mit 1.0 bewertet, deshalb erhalten wir $\hat{Q}(b,u) = 0 + 0.8 \cdot 1 = 0.8$. Im Feld e wird daraufhin die Aktion r gewählt und die Q-Werte ändern sich nicht.

a 0.0 0.0	0.0 b 0.0 0.8	0.0 c 0.0
0.0 d 0.0	0.0 0.0 e 1.0	0.0 0.0 f

Ein weiterer Durchlauf sei folgenden Weg gegangen : $d \rightarrow e \rightarrow f$ (wobei die Wahl der nächsten Aktion im Feld e eindeutig war).

a 0.0 0.0	0.0 b 0.0 0.8	0.0 c 0.0
0.0 d 0.8	0.0 0.0 e 1.0	0.0 0.0 f

Weitere Durchläufe ergaben folgende Wege : $c \rightarrow f$ und $a \rightarrow b \rightarrow e \rightarrow f$

a 0.64 0.0	0.0 b 0.0 0.8	0.0 c 1.0
0.0 d 0.8	0.0 0.0 e 1.0	0.0 0.0 f

In weiteren Durchläufen finden keine weiteren Änderungen an den Q-Werten mehr statt. Insgesamt wurde eine (pseudo-)optimale Policy gefunden, die unten zu sehen ist. Beachten Sie, dass die ermittelten Q-Werte nicht immer optimal sein müssen. Die Konvergenz von Q-Learning an die optimale Q-Funktion gilt im Allgemeinen nur, wenn jedes Zustands-Aktions Paar beliebig oft besucht wird.

$\rightarrow$	$\downarrow$	$\downarrow$
$\rightarrow$	$\rightarrow$	

Ein Beispiel für Simulationssequenzen, so dass Q-Learning mit einer minimalen Anzahl an Updates konvergiert: $c \rightarrow f, e \rightarrow f, b \rightarrow c \rightarrow f, d \rightarrow e \rightarrow f, a \rightarrow b \rightarrow c \rightarrow f$