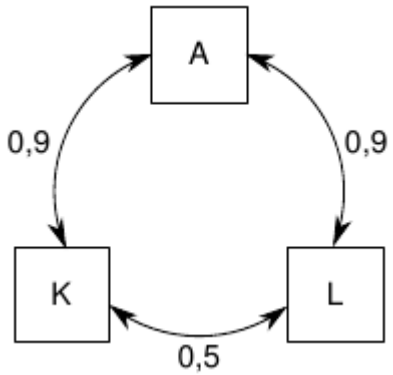
Übungen zur Vorlesung
Grundlagen der Künstlichen Intelligenz
Sommersemester 2024

Institut für Informatik 4
Priv.-Doz. Dr. V. Steinhage
Friedrich-Hirzebruch-Allee 8
53115 Bonn
Email: steinhage@cs.uni-bonn.de
WWW: http://net.cs.uni-bonn.de/ivs/

Blatt 7 (5 Punkte + 2 Bonuspunkte)

Abgabe durch Hochladen (nur PDF-Format bzw. Python-Code) auf der eCampus-Seite bis **Sonntag, 02.06.2024, 12:00 Uhr**, in Gruppen von 3 Personen.

Aufgabe 7.1: MDP: Entwurf und Anwendung von *Value Iteration* $(\mathbf{2+2=4})$



Ihr Agent startet in A. In K liegt ein Schlüssel, den Ihr Agent bei einem Besuch mitnimmt. Dieser Schlüssel passt auf das Schloss in L. Gelangt Ihr Agent mit dem Schlüssel dorthin, schließt er das Schloss auf und findet dahinter eine Belohnung von $+1$. Mit Erreichen von L terminiert der Agent, sofern er in Besitz des Schlüssels ist. Alle anderen Schritte verursachen jeweils Kosten von $-0,1$. Der Wechsel zwischen zwei Orten gelingt nur mit einer gewissen Wahrscheinlichkeit, die Sie der obigen Zeichnung entnehmen können, im anderen Fall bleibt der Agent im gleichen Ort.

a) Finden Sie eine Modellierung dieses Problems, in der diese Nutzenfunktion separierbar ist und der Nutzen eines Zustands nicht mehr von der vergangenen Historie abhängt. Stellen Sie diese Modellierung als Graph dar, bei dem die Knoten die Zustände und deren Nutzenwerte enthalten. Es sollte also fünf Knoten geben mit den Zuständen $s_0 = (A, -S)$, $s_1 = (K, S)$, $s_2 = (L, -S)$, $s_3 = (A, S)$ und $s_4 = (L, S)$, wobei S beschreibt, dass der Agent den Schlüssel besitzt. Verbinden Sie die Zustandsknoten *vollständig* durch *alle möglichen* Zustandsübergange bzw. Aktionen als gerichtete Kanten. Schreiben Sie die entsprechenden Übergangswahrscheinlichkeiten an alle Kanten.

b) Führen Sie den Algorithmus VALUE ITERATION für *eine* Iteration auf ihrem Entwurf durch: berechnen Sie so die Nutzen $U_1(s_0)$, $U_1(s_1)$, $U_1(s_2)$, $U_1(s_3)$, $U_1(s_4)$ und geben Sie die *policy*$_1$ der Zustände s_0,...,s_3 dieser ersten Iteration an.
Hinweis 1: Es gilt $U_0(s_0) = U_0(s_1) = U_0(s_2) = U_0(s_3) = -0.1$ und $U_0(s_4) = +1$.
Hinweis 2: Sollte die Value-Iteration für einen Zustand mehrere Nachfolgezustände mit max. erwarteten Nutzen ergeben, wird die Policy für diesen Zustand enstpr. viele Aktionen vorschlagen (z.B.: *gehe_zu_L* oder *gehe_zu_A*).

Aufgabe 7.2: MDP: Alternative Reward-Funktion **(1 Punkt + 2 Bonuspunkte)**

Man könnte ein MDP auch mit einer Reward-Funktion $R(s, a)$ für Zustände s und Aktionen a formulieren, so dass der Reward also sowohl vom Zustand s als auch von der in diesem ausgeführten Aktion a abhängig ist. Damit werden bestimmte Aktionen in bestimmten Zuständen also unabhängig von Ausgang besser oder schlechter bewertet.

1. Schreiben Sie die Bellmann-Gleichung $U(s) = ...$ für diese Formulierung mit einer Reward-Funktion $R(s, a)$.

2. Bonusaufgabe: Zeigen Sie, wie ein MDP mit einer Reward-Funktion $R(s, a)$ und Transitionsmodell M in ein anderes MDP mit einer Reward-Funktion $R'(s)$ und Transitionsmodell M' umgewandelt werden kann, so dass optimale Strategien im neuen MDP mit $R'(s)$ genau den optimalen Strategien im ursprünglichen MDP $R(s, a)$ entsprechen. Hinweis: Sie mögen Zustände $post(s, a)$ für Zustände s und Aktionen a zur Umformulierung zur Hilfe nehmen.