Équations de Bellman pour la valeur optimale

- Les **équations de Bellman** nous donnent une condition qui est garantie par la valeur V^* des plans optimaux

$$V^*(s) = R(s) + \max_{a} \ \gamma \sum_{s' \in S} P(s' | s,a) \ V^*(s') \quad \forall \ s \in S$$

- Deux algorithmes différents pour le calcul du plan optimal:
  - **itération par valeurs** (***value iteration***)
  - **itération par politiques** (***policy iteration***)

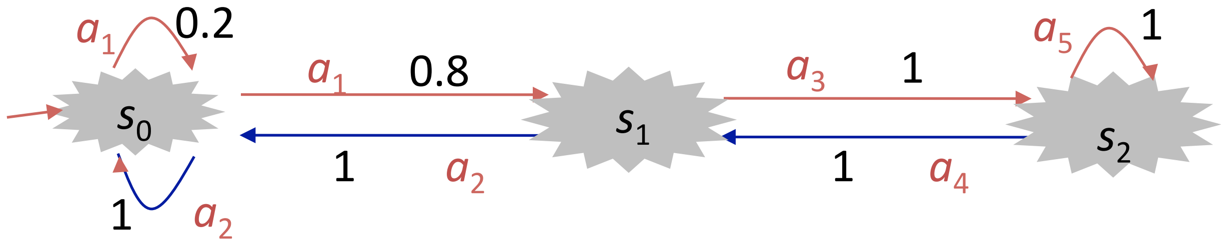
Hugo Larochelle et Froduald Kabanza

Algorithme *policy iteration*

1. Choisir un plan arbitraire π'
2. Répéter jusqu'à ce que le plan ne change pas ($\pi = \pi'$) :
   I. $\pi \leftarrow \pi'$
   II. pour tout *s* dans *S*, calculer $V(\pi,s)$ en résolvant le système de $|S|$ équations et $|S|$ inconnues
   $$V(\pi,s) = R(s) + \gamma \, \Sigma_{s' \in S} P(s'|s, \pi(s)) \, V(\pi,s')$$
   III. pour tout *s* dans *S*, s'il existe une action *a* telle que
   $$[\, R(s) + \gamma \, \Sigma_{s' \in S} P(s'|s,a) \, V(\pi,s') \,] > V(\pi,s)$$
   alors $\pi'(s) := a$ sinon $\pi'(s) := \pi(s)$
3. Retourne π

Hugo Larochelle et Froduald Kabanza

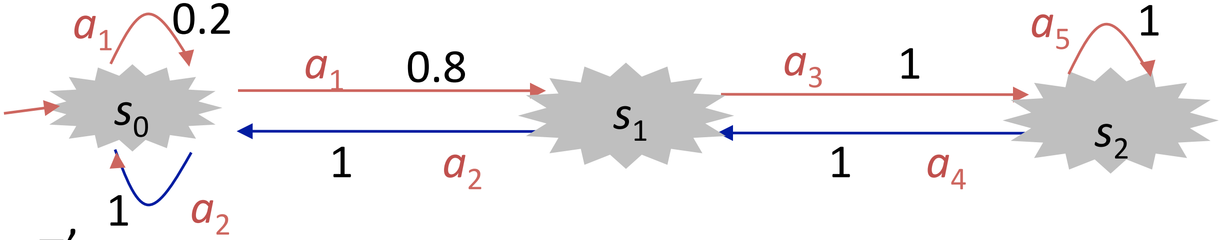
Policy iteration: initialisation



- Plan initial choisi arbitrairement:

$$\pi' = \{ s_0 \rightarrow a_2, \\ s_1 \rightarrow a_2, \\ s_2 \rightarrow a_4 \}$$

Policy iteration: itération #1



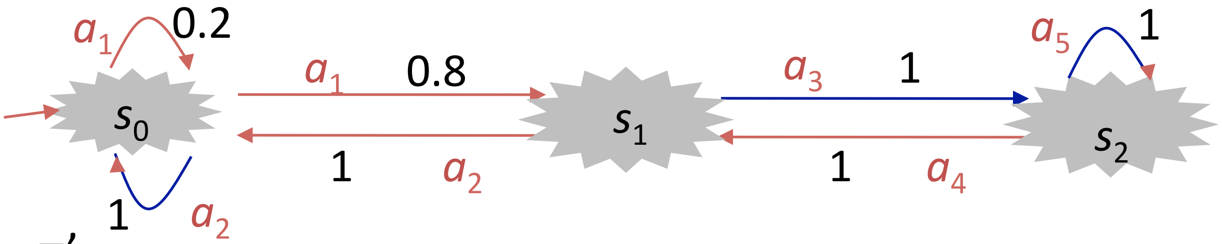
I. $\pi \leftarrow \pi'$

II. Équations: $v_0=0+0.5*(1*v_0)$;
$v_1=0+0.5*(1*v_0)$;
$v_2=1+0.5*(1*v_1)$

Solution: $v_0=0$, $v_1=0$, $v_2=1$

III. $s_0 \rightarrow a_1$: $0+0.5*(0.2*0+0.8*0)=0$; ne change pas
$s_1 \rightarrow a_3$: $0+0.5*(1*1)=0.5 > 0$; **change**
$s_2 \rightarrow a_5$: $1+0.5*(1*1)=1.5 > 1$; **change**
$\pi' = \{ s_0 \rightarrow a_2 , s_1 \rightarrow a_3 , s_2 \rightarrow a_5 \}$

Policy iteration: itération #2

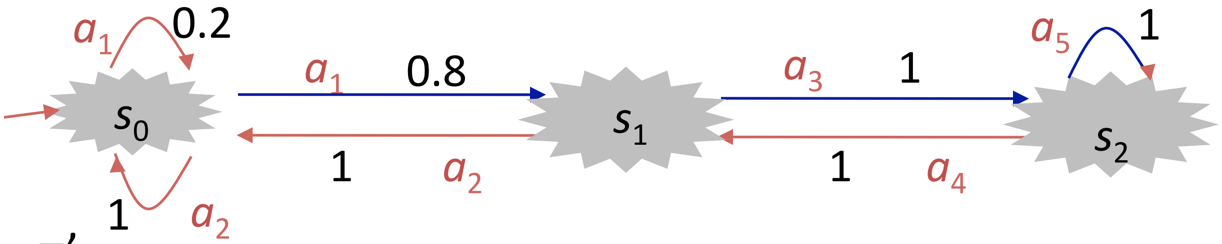


I. $\pi \leftarrow \pi'$

II. Équations: $v_0=0+0.5*(1*v_0);$
$v_1=0+0.5*(1*v_2);$
$v_2=1+0.5*(1*v_2)$
Solution: $v_0=0$, $v_1=1$, $v_2=2$

III. $s_0 \rightarrow a_1$: $0+0.5(0.2*0+0.8*1)=0.4 > 0;$ **change**
$s_1 \rightarrow a_2$: $0+0.5(1*0)=0 < 1;$ ne change pas
$s_2 \rightarrow a_4$: $1+0.5(1*1)=1.5 < 2;$ ne change pas
$\pi' = \{ s_0 \rightarrow a_1 , s_1 \rightarrow a_3 , s_2 \rightarrow a_5 \}$

Policy iteration: itération #3



I. $\pi \leftarrow \pi'$

II. Équations: v_0=0+0.5*(0.2*v_0+0.8*v_1);
v_1=0+0.5*(1*v_2);
v_2=1+0.5*(1*v_2)

Solution: v_0=4/9, v_1=1, v_2=2

III. $s_0 \rightarrow a_{2:}$ 0+0.5(1*0.4)=0.2 < 4/9; ne change pas
$s_1 \rightarrow a_{2:}$ 0+0.5(1*0.4)=0.2 < 1; ne change pas
$s_2 \rightarrow a_{4:}$ 1+0.5(1*1)=1.5 < 2; ne change pas
$\pi' = \{ s_0 \rightarrow a_1 , s_1 \rightarrow a_3 , s_2 \rightarrow a_5 \}$, c-à-d. π

Solution trouvée

Rappel: systèmes d'équations linéaires

- Soit le système d'équations:
  $v_0 = 0 + 0.5 * (0.2*v_0+0.8*v_1);$
  $v_1 = 0 + 0.5 * (1*v_2);$
  $v_2 = 1 + 0.5 * (1*v_2)$
- En mettant toutes les variables à droite, on peut l'écrire sous la forme:
  $0 = -0.9\ v_0 + 0.4\ v_1$ (1)
  $0 = -v_1 + 0.5\ v_2$ (2)
  $-1 = -0.5\ v_2$ (3)
- De l'équation (3), on conclut que $v_2 = -1 / -0.5 = 2$
- De l'équation (2), on conclut que $v_1 = 0.5\ v_2 = 1$
- De l'équation (1), on conclut que $v_0 = 0.4\ v_1 / 0.9 = 4/9$

Rappel: systèmes d'équations linéaires

- Soit le système d'équations:
  $v_0 = 0 + 0.5 * (0.2*v_0+0.8*v_1);$
  $v_1 = 0 + 0.5 * (1*v_2);$
  $v_2 = 1 + 0.5 * (1*v_2)$
- En mettant toutes les variables à droite, on peut l'écrire sous la forme:
  $0 = -0.9\, v_0 + 0.4\, v_1 \qquad (1)$
  $0 = -v_1 + 0.5\, v_2 \qquad (2)$
  $-1 = -0.5\, v_2 \qquad (3)$
- Approche alternative: on écrit sous forme matricielle b = A v, où

$$A = \begin{pmatrix} -0.9 & 0.4 & 0 \\ 0 & -1 & 0.5 \\ 0 & 0 & -0.5 \end{pmatrix} \quad b = \begin{pmatrix} 0 \\ 0 \\ -1 \end{pmatrix} \quad v = \begin{pmatrix} v_0 \\ v_1 \\ v_2 \end{pmatrix}$$

Rappel: systèmes d'équations linéaires

- Suffit alors d'inverser A pour obtenir v = A^{-1} b
  - on peut utiliser une librairie d'algèbre linéaire (ex.: Numpy en Python):

```
>>> A = numpy.array([[-0.9,0.4,0],[0,-1,0.5],[0,0,-0.5]])
>>> b = numpy.array([0,0,-1])
>>> Ainv = numpy.linalg.inv(A)
>>> v = numpy.dot(Ainv,b)
>>> print v
[ 0.44444444  1.          2.        ]
```

$$A = \begin{pmatrix} -0.9 & 0.4 & 0 \\ 0 & -1 & 0.5 \\ 0 & 0 & -0.5 \end{pmatrix} \quad b = \begin{pmatrix} 0 \\ 0 \\ -1 \end{pmatrix} \quad v = \begin{pmatrix} v_0 \\ v_1 \\ v_2 \end{pmatrix}$$