Contents

1 Probability recap

Definition 1.1. Marginal distribution
Given a joint distribution over two random variables X and Y, the *marginal distribution* of X is defined as

$$P_X(x) = \sum_y P_{X,Y}(x, y) \tag{Marginal distribution}$$

A nice reference for this is the wikipedia page on Marginal distribution.

Definition 1.2. Expected value
Considering a random variable X with a finite list $x_1, \ldots, x_k$ of possible outcome values and corresponding probabilities $p_1, \ldots, p_k$, the *expected value* $\mathbb{E}[X]$ of X is defined as

$$\mathbb{E}[X] = \sum_{i=1}^{k} p_i x_i \tag{Expected value}$$

Similarly, for a function $f(X)$ of the random variable X, the expected value is defined as

$$\mathbb{E}_{x \sim P}[f(X)] = \sum_x P(X = x) f(x) \tag{Expected value of function}$$

Here the notation $x \sim P$ indicates that the variable x is drawn from the distribution P. Generalizing this to multivariate distributions, for two random variables X and Y with joint distribution $P_{X,Y}$, the expected value of a function $f(X, Y)$ is defined as

$$\mathbb{E}_{(x,y) \sim P}[f(X, Y)] = \sum_x \sum_y P_{X,Y}(x, y) f(x, y) \tag{Expected value of multivariate function}$$

Definition 1.3. Conditional expectation
Given two random variables X and Y, the *conditional expectation* of X given $Y = y$ is defined as

$$\mathbb{E}[X \mid Y = y] = \sum_x x \cdot P_{X|Y}(x \mid y) \tag{Conditional expectation}$$

Definition 1.4. Conditional distribution
Given a joint distribution over two random variables X and Y, the *conditional distribution* of X given $Y = y$ is defined as

$$P_{X|Y}(x \mid y) = \frac{P_{X,Y}(x, y)}{P_Y(y)} \tag{Conditional distribution}$$

Equivalently, we can expand the denominator using the marginal distribution:

$$P_{X|Y}(x \mid y) = \frac{P_{X,Y}(x, y)}{\sum_{x'} P_{X,Y}(x', y)} \tag{Conditional distribution (using marginal)}$$

Definition 1.5. Chain rule of probability
Given a sequence of random variables $X_1, X_2, \ldots, X_n$, the *chain rule of probability* states that the joint distribution can be decomposed as

$$P(X_1, X_2, \ldots, X_n) = P(X_1)P(X_2 \mid X_1)P(X_3 \mid X_1, X_2) \ldots P(X_n \mid X_1, X_2, \ldots, X_{n-1}) \tag{Chain rule of probability}$$

Equivalently we can express this using product notation as

$$P(X_1, X_2, \ldots, X_n) = \prod_{i=1}^{n} P(X_i \mid X_1, X_2, \ldots, X_{i-1}) \tag{Chain rule of probability (product notation)}$$

2 Finite Markov Decision Processes

Definition 2.1. Agent-Environment Interface
We consider some agent interacting with an environment in discrete time steps $t = 0, 1, 2, \ldots$. At each time step t, the agent

1. *State* - observes some representation of the environment's state $S_t \in \mathcal{S}$
2. *Action* - chooses an action $A_t \in \mathcal{A}$
3. *Reward* - receives a reward $R_{t+1} \in \mathbb{R}$, and the environment transitions to a new state S_{t+1}

Hence we yield a sequence or *trajectory* of states, actions, and rewards:

$$S_0, A_0, R_1, S_1, A_1, R_2, S_2, A_2, R_3, \ldots$$

Definition 2.2. Configuration probability
The probability of a particular state s' and reward r occurring at time $t+1$, given the current state s and action a at time t, is given by the *configuration probability*:

$$p(s', r \mid s, a) = P(S_{t+1} = s', R_{t+1} = r \mid S_t = s, A_t = a) \tag{Configuration probability 1}$$

or equivalently at a time t as

$$p(s', r \mid s, a) = P(S_t = s', R_t = r \mid S_{t-1} = s, A_{t-1} = a) \tag{Configuration probability 2}$$

The distribution p here fully characterises the distribution for each choice of state s and action a.

$$\sum_{s' \in \mathcal{S}} \sum_{r \in \mathbb{R}} p(s', r \mid s, a) = 1 \quad \forall s \in \mathcal{S}, a \in \mathcal{A}_s$$

Definition 2.3. State transition probability
The probability of a new state s' occuring at a time t given the current state s and action a at time $t-1$ is given by the *state transition probability*:

$$\begin{aligned} p(s' \mid s, a) &: \text{State} \times \text{State} \times \text{Action} \to [0, 1] \\ &= P(S_t = s' \mid S_{t-1} = s, A_{t-1} = a) \\ &= \sum_{r \in \mathbb{R}} p(s', r \mid s, a) \end{aligned} \tag{State transition probability}$$

Note that this is simply the marginal distribution over the configuration probability $p(s', r \mid s, a)$. In other words, we sum out the reward variable r to get the probability of transitioning to state s' given state s and action a.

Definition 2.4. State reward probability
The probability of receiving a reward r at time t given the current state s and action a at time $t-1$ is given by the *state reward probability*:

$$\begin{aligned} p(r \mid s, a) &: \text{State} \times \text{Action} \times \mathbb{R} \to [0, 1] \\ &= P(R_t = r \mid S_{t-1} = s, A_{t-1} = a) \\ &= \sum_{s' \in \mathcal{S}} p(s', r \mid s, a) \end{aligned} \tag{State reward probability}$$

Definition 2.5. Expected configuration reward
The expected reward when taking action a in state s is given by the *expected configuration reward*:

$$\begin{aligned} r(s, a) &: \text{State} \times \text{Action} \to \mathbb{R} \\ &= \mathbb{E}[R_t \mid S_{t-1} = s, A_{t-1} = a] \\ &= \sum_{r} r \cdot p(r \mid s, a) \end{aligned} \tag{Expected configuration reward}$$

Definition 2.6. State-action-next expected reward

The expected reward when taking action a in state s and transitioning to state s' is given by the *state-action-next expected reward*:

$$\begin{aligned} r(s, a, s') &: \text{State} \times \text{Action} \times \text{State} \to \mathbb{R} \\ &= \mathbb{E}[R_t \mid S_{t-1} = s, A_{t-1} = a, S_t = s'] \\ &= \sum_{r \in \mathbb{R}} r \cdot \frac{p(s', r \mid s, a)}{p(s' \mid s, a)} \end{aligned} \qquad \text{(SAN expected reward (1))}$$

As an alternative form, we can also use the chain rule of probability to express this as

$$r(s, a, s') = \sum_{r \in \mathbb{R}} r \cdot p(r \mid s, a, s') \qquad \text{(SAN expected reward (2))}$$

We can again provide some intuition here, so expanding the marginalization in the denominator, we have

$$r(s, a, s') = \sum_{r \in \mathbb{R}} r \cdot \frac{p(s', r \mid s, a)}{\sum_{r'} p(s', r' \mid s, a)}$$

Simplify this by just focusing on s' and r we have

$$r(s') = \sum_{r \in \mathbb{R}} r \cdot P(R_t = r \mid S_t = s')$$

so we can see that all we are doing here is just computing the expected value of the reward variable conditioned on the next state s' (ignoring the previous state and action).

Definition 2.7. Simple return

In general, we want to maximize the *expected return* G_t, in the simple case we define it just as the sum of the rewards from time step $t + 1$ to some terminal time step T:

$$G_t = R_{t+1} + R_{t+2} + \ldots + R_T \qquad \text{(Simple return)}$$

Definition 2.8. Finite horizon

A *finite horizon* MDP is one where there exists a time step T such that for all $t \geqslant T$, the state is terminal, i.e., $S_t = s_{\text{terminal}}$ and no further rewards are obtained, i.e., $R_t = 0$. Thus the return at any time step t can be expressed as

$$G_t = \sum_{k=t+1}^{T} R_k \qquad \text{(Simple return (finite horizon))}$$

Definition 2.9. Discounted return

Commonly we introduce a notion of *discounted return* to prioritise immediate rewards over distant future rewards. Given a discount factor $\gamma \in [0, 1)$, the discounted return G_t at time step t is defined as

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \ldots = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \qquad \text{(Discounted return)}$$

Here with the discount factor

- if $\gamma = 0$, the agent is *myopic* and only cares about immediate rewards
- if $\gamma \to 1$, the agent values future rewards almost as much as immediate rewards

if the reward is fixed to +1 at each time step, then the discounted return becomes a geometric series:

$$G_t = 1 + \gamma + \gamma^2 + \ldots = \frac{1}{1 - \gamma}$$

Another useful form of the discounted return is to express it in terms of finite horizon T:

$$G_t = \sum_{k=t+1}^{T} \gamma^{k-t-1} R_k \qquad \text{(Discounted return (finite horizon))}$$

Definition 2.10. Discounted return recursion
Discounted returns at each successive time steps can be related via the following recursion:

$$G_t = R_{t+1} + \gamma G_{t+1} \tag{Discounted return recursion}$$

Derived as follows:

$$\begin{aligned} G_t &= R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \ldots \\ &= R_{t+1} + \gamma(R_{t+2} + \gamma R_{t+3} + \ldots) \\ &= R_{t+1} + \gamma G_{t+1} \end{aligned}$$

Definition 2.11. Policy function
A policy π is a mapping from states to a probability distribution over actions:

$$\begin{aligned} \pi(a \mid s) &: \text{Action} \times \text{State} \to [0, 1] \\ &= P(A_t = a \mid S_t = s) \end{aligned} \tag{Policy function}$$

where $\pi(a \mid s)$ gives the probability of taking action a when in state s.

Definition 2.12. Value function
The value function $v_\pi(s)$ of a state s under a policy π is defined as the expected return when starting in state s and following policy π thereafter:

$$v_\pi(s) = \mathbb{E}_\pi[G_t \mid S_t = s] \tag{Value function}$$

In other words we can see that its expressed as the conditional expectation of the return G_t given that we start in state s at time t and follow policy π. Expending this using the definition of discounted return recursion, we have

$$v_\pi(s) = \mathbb{E}_\pi\left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \mid S_t = s\right]$$

Using the recursive form of the discounted return, we can also express this as

$$\begin{aligned} v_\pi(s) &= \mathbb{E}_\pi[G_t \mid S_t = s] \\ &= \mathbb{E}_\pi[R_{t+1} + \gamma G_{t+1} \mid S_t = s] \end{aligned} \tag{Recursive value function}$$

Fully expanding the recursive form, we have

$$\begin{aligned} v_\pi(s) &= \mathbb{E}_\pi[G_t \mid S_t = s] \\ &= \mathbb{E}_\pi[R_{t+1} + \gamma G_{t+1} \mid S_t = s] \\ &= \sum_a \pi(a \mid s) \sum_{s'} \sum_r p(s', r \mid s, a)\, [r + \gamma\, \mathbb{E}_\pi[G_{t+1} \mid S_{t+1} = s']] \\ &= \sum_a \pi(a \mid s) \sum_{s'} \sum_r p(s', r \mid s, a)\, [r + \gamma v_\pi(s')] \end{aligned} \tag{Expanded recursive value function}$$

The book explains this derivation a bit badly, its basically just an expansion of expected values sampling over actions, next states, and rewards. So starting out lets consider some function $f(A)$ defined over actions A. Then we can express the expected value of this function under policy π as

$$\mathbb{E}_{A\sim\pi(-|s)}[f(A)] = \sum_a \pi(a \mid s) f(a)$$

For the inner term we are talking about the expected value sampling over next states and rewards, so we can express this as

$$\begin{aligned}\mathbb{E}_{(S',R)\sim p(-|s,a)}[g(S',R)] &= \sum_{s'}\sum_{r} p(s',r \mid s,a)g(s',r) \\ &= \sum_{s',r} p(s',r \mid s,a)g(s',r)\end{aligned}$$

where $g(S',R)$ is some function defined over next states and rewards. Now substituting in our actual functions, we have

$$f(A) = \mathbb{E}_{\pi}[R_{t+1} + \gamma G_{t+1} \mid S_t = s, A_t = A]$$

which is the expected return given state s and action A, and

$$g(S',R) = R + \gamma\,\mathbb{E}_{\pi}[G_{t+1} \mid S_{t+1} = S']$$

which is the expected return given next state S' and reward R. Plugging these into the expected value expansions above gives us the full expansion as shown. An alternative representation which makes more sense in terms of figuring out the derivation would be as follows

$$v_{\pi}(s) = \mathbb{E}_{A\sim\pi(-|s),(S',R)\sim p(-|s,A)}[R + \gamma v_{\pi}(S')] \qquad \text{(Alternative recursive value function)}$$

So the value function at state s is the expected value over actions sampled from policy π and next states and rewards sampled from the configuration probability, of the reward plus the discounted value of the next state.

Definition 2.13. Action-value function
The action-value function $q_{\pi}(s,a)$ of a state-action pair (s,a) under a policy π is defined as the expected return when starting in state s, taking action a, and thereafter following policy π:

$$q_{\pi}(s,a) = \mathbb{E}_{\pi}[G_t \mid S_t = s, A_t = a] \qquad \text{(Action-value function)}$$

Similar to the value function, we can expand this using the definition of discounted return recursion:

$$q_{\pi}(s,a) = \mathbb{E}_{\pi}\left[\sum_{k=0}^{\infty}\gamma^k R_{t+k+1} \mid S_t = s, A_t = a\right]$$

Definition 2.14. Optimal policy & state-value function
A policy π is defined to be better than or equal to another policy π' if its value function is greater than or equal to that of π' for all states $s \in \mathcal{S}$:

$$\pi \geqslant \pi' \iff v_{\pi}(s) \geqslant v_{\pi'}(s) \quad \forall s \in \mathcal{S} \qquad \text{(Policy comparison)}$$

An *optimal policy* π^* is one that is better than or equal to all other policies:

$$\pi^* \geqslant \pi \quad \forall \pi \qquad \text{(Optimal policy)}$$

The corresponding *optimal state-value function* $v_*(s)$ is defined as the value function under the optimal policy:

$$v_*(s) = v_{\pi^*}(s) = \max_{\pi} v_{\pi}(s) \quad \forall s \in \mathcal{S} \qquad \text{(Optimal state-value function)}$$

Definition 2.15. Optimal action-value function
The *optimal action-value function* $q_*(s,a)$ is defined as the action-value function under the optimal policy:

$$q_*(s,a) = q_{\pi^*}(s,a) = \max_{\pi} q_{\pi}(s,a) \quad \forall s \in \mathcal{S}, a \in \mathcal{A}_s \qquad \text{(Optimal action-value function 1)}$$

Define in terms of the optimal state-value function, we have

$$q_*(s,a) = \mathbb{E}[R_{t+1} + \gamma v_*(S_{t+1}) \mid S_t = s, A_t = a] \qquad \text{(Optimal action-value function 2)}$$

Definition 2.16. Bellman optimality equation
The Bellman equation for the optimal state-value function is given by

$$v_*(s) = \max_a \sum_{(s',r)} p(s', r \mid s, a)\,[r + \gamma v_*(s')] \tag{Bellman optimality equation}$$

It's derivation is based on the idea that the state value function under the optimal policy must choose the action that maximises the expected return, hence we take the max over actions. Then we expand the expected return using the configuration probability and the recursive form of the value function. So the full derivation

$$\begin{aligned} v_*(s) &= \max_{a \in A_s} q_*(s, a) \\ &= \max_{a \in A_s} \mathbb{E}[R_{t+1} + \gamma v_*(S_{t+1}) \mid S_t = s, A_t = a] \\ &= \max_{a \in A_s} \sum_{(s',r)} p(s', r \mid s, a)\,[r + \gamma v_*(s')] \end{aligned}$$

Another variant of this equation is to marginalize out the reward variable r to express it in terms of the state transition probability and expected configuration reward:

$$v_*(s) = \max_a \sum_{s'} p(s' \mid s, a)\,[r(s, a, s') + \gamma v_*(s')] \tag{Bellman optimality equation (variant)}$$

If it seems a bit confusing how we get there we can do it step by step, an important equality to note is that

$$p(s', r \mid s, a) = p(s' \mid s, a) \cdot p(r \mid s, a, s') \tag{Chain rule application}$$

So starting from the original Bellman optimality equation, we have

$$\begin{aligned} v_*(s) &= \max_a \sum_{s'} \sum_r p(s', r \mid s, a)\,[r + \gamma v_*(s')] \\ &= \max_a \sum_{s'} \sum_r p(s' \mid s, a) p(r \mid s, a, s')\,[r + \gamma v_*(s')] && \text{(chain rule)} \\ &= \max_a \sum_{s'} p(s' \mid s, a) \left[\sum_r r \cdot p(r \mid s, a, s') + \gamma v_*(s') \right] && \text{(distributing)} \\ &= \max_a \sum_{s'} p(s' \mid s, a)\,[r(s, a, s') + \gamma v_*(s')] \end{aligned}$$

important to note when we distributed the sums we had

$$\sum_r p(r \mid s, a, s') = 1$$

hence why the $\gamma v_*(s')$ term remains unchanged.

Definition 2.17. Finite horizon bellman optimality
If we consider a finite horizon MDP lasting for T time steps, then we can define the Bellman optimality equation at each time step t with $t = 0, 1, \ldots, T - 1$ as

$$v_*^t(s) = \max_a \sum_{s'} p(s' \mid s, a)\left[r(s, a, s') + \gamma v_*^{t+1}(s')\right] \tag{Finite horizon Bellman optimality}$$

Definition 2.18. FHMDP Matrix form
We define the following matrix and vector representations for the finite horizon MDP Bellman optimality equation:

$$\begin{aligned} \mathbf{v}_t \in \mathbb{R}^{|\mathcal{S}|}, &\quad [\mathbf{v}_t]_s = v_*^t(s) && \text{(Value vector)} \\ P_a \in \mathbb{R}^{|\mathcal{S}| \times |\mathcal{S}|}, &\quad [P_a]_{s,s'} = p(s' \mid s, a) && \text{(State transition matrix)} \\ \mathbf{r}_a \in \mathbb{R}^{|\mathcal{S}|}, &\quad [\mathbf{r}_a]_s = \sum_{s'} p(s' \mid s, a) r(s, a, s') && \text{(Expected reward vector)} \end{aligned}$$

Our action-value vector can then be expressed as

$$\mathbf{q}_a^t = \mathbf{r}_a + \gamma P_a \mathbf{v}_{t+1} \tag{Action-value vector}$$

and the Bellman optimality equation in matrix form is given by

$$\mathbf{v}_t = \max_a \mathbf{q}_a^t = \max_a (\mathbf{r}_a + \gamma P_a \mathbf{v}_{t+1}) \tag{FHMDP Bellman optimality (matrix form)}$$

Equivalently in index form we have

$$[\mathbf{v}_t]_s = v_*^t(s) = \max_a \left([\mathbf{r}_a]_s + \gamma \sum_{s'} [P_a]_{s,s'} [\mathbf{v}_{t+1}]_{s'} \right) \tag{FHMDP Bellman optimality (index form)}$$

Expressed in a less compact form we have

$$\max_a \left(\begin{bmatrix} \sum_{s'} p(s' \mid s_0, a) r(s, a, s') \\ \sum_{s'} p(s' \mid s_1, a) r(s, a, s') \\ \vdots \\ \sum_{s'} p(s' \mid s_n, a) r(s, a, s') \end{bmatrix} + \gamma \begin{bmatrix} p(s_0' \mid s_0, a) & p(s_1' \mid s_0, a) & \dots & p(s_n' \mid s_0, a) \\ p(s_0' \mid s_1, a) & p(s_1' \mid s_1, a) & & p(s_n' \mid s_1, a) \\ \vdots & & \ddots & \vdots \\ p(s_0' \mid s_n, a) & p(s_1' \mid s_n, a) & \dots & p(s_n' \mid s_n, a) \end{bmatrix} \begin{bmatrix} v_*^{t+1}(s_0) \\ v_*^{t+1}(s_1) \\ \vdots \\ v_*^{t+1}(s_n) \end{bmatrix} \right)$$

Here we can see nicely that for each state s_i we are computing the expected reward plus the discounted expected value of the next state, and then taking the max over actions. For example if we take some $t = k$ and look at the first element of the vector, we have

$$v_*^k(s_0) = \max_a \left(\sum_{s'} p(s' \mid s_0, a) r(s_0, a, s') + \gamma \sum_{s'} p(s' \mid s_0, a) v_*^{k+1}(s') \right)$$

which matches the Bellman optimality equation as expected (before factoring out the $p(s' \mid s, a)$ term).

Definition 2.19. Tensor contraction
A *tensor contraction* is an operation that reduces the order of a tensor by summing over one or more of its indices. In general we would say given a tensor U of order $s + v$ and a tensor V of order $v + t$, their contraction over the v shared indices or *mode* results in a new tensor W of order $s + t$ defined as

$$W_{i_1,\dots,i_s,j_1,\dots,j_t} = \sum_{k_1,\dots,k_v} U_{i_1,\dots,i_s,k_1,\dots,k_v} V_{k_1,\dots,k_v,j_1,\dots,j_t} \tag{Tensor contraction}$$

where the summation is over all possible values of the shared indices $k_1, \dots, k_v$. We can also introduce an alternative matrix style notation for tensor contractions

$$W = U \times_v V \tag{Tensor contraction (matrix style)}$$

To provide a more concrete - albiet somewhat contrived - example, let's consider a tensor T of shape $3 \times 3 \times 3$ and vector $\mathbf{v}$ of shape 3. Now let's asses the ways; i.e. types of tensor contractions; we can perform between these two objects. For the sake of simplicitly let's just consider a the first slice of our tensor T so we have

$$T = \begin{bmatrix} t_{111} & t_{112} & t_{113} \\ t_{121} & t_{122} & t_{123} \\ t_{131} & t_{132} & t_{133} \end{bmatrix}, \quad \mathbf{v} = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix}$$

Now the classical matrix-vector product is a contraction over the third mode of the tensor and the first mode of the vector, yielding a vector of shape 3:

$$\mathbf{w} = T \times_k \mathbf{v} = \begin{bmatrix} t_{i11}v_1 + t_{i12}v_2 + t_{i13}v_3 \\ t_{i21}v_1 + t_{i22}v_2 + t_{i23}v_3 \\ t_{i31}v_1 + t_{i32}v_2 + t_{i33}v_3 \end{bmatrix} \quad \forall i \in \{1, 2, 3\}$$

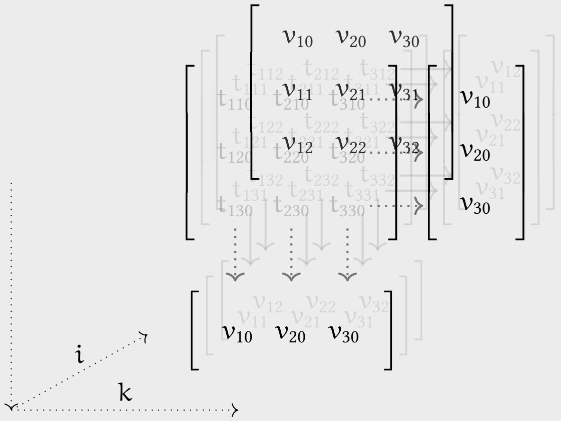
if we define a cell as being indexed by (i, j, k) then we can equivalently understand this as producing a weighted sum over the k axis. An intuitive way of viewing this is if you imagine shining a light along the k axis, so from left to right, then the resulting shadow - in the 2d case - on the jth plane is the resulting vector made up of the weighted sums for each row. We can now ask: *what if we shine the light along the* j *axis instead?* In this case we would be contracting over the second mode of the tensor and the first mode of the vector, yielding in our simple case a vector of shape 3 as follows:

$$\mathbf{u} = T \times_j \mathbf{v} = \begin{bmatrix} t_{i11}v_1 + t_{i21}v_2 + t_{i31}v_3 \\ t_{i12}v_1 + t_{i22}v_2 + t_{i32}v_3 \\ t_{i13}v_1 + t_{i23}v_2 + t_{i33}v_3 \end{bmatrix} \quad \forall i \in \{1, 2, 3\}$$

So we can see here that intuitively that each element corresponds to the weighted sum taken from top to bottom i.e. along the j axis. If we then go back and consider our original tensor of shape $3 \times 3 \times 3$ we can also consider contracting over the first mode of the tensor and the first mode of the vector, intuitively this means shining a light essentially facing the tensor, yielding a vector of shape 3 as follows:

$$\mathbf{z} = T \times_i \mathbf{v} = \begin{bmatrix} t_{1j1}v_1 + t_{2j1}v_2 + t_{3j1}v_3 \\ t_{1j2}v_1 + t_{2j2}v_2 + t_{3j2}v_3 \\ t_{1j3}v_1 + t_{2j3}v_2 + t_{3j3}v_3 \end{bmatrix} \quad \forall j \in \{1, 2, 3\}$$

To visualise this better we can use tikz to draw out the tensor and vector contractions with some arrows to indicate the direction of the contraction. The faded matrices in the background represent the original tensor and vector, while the arrows indicate the direction of contraction.



v_{10} v_{20} v_{30}
v_{11} v_{21} v_{31}
v_{12} v_{22} v_{32}
t_{110} t_{210} t_{310}
t_{120} t_{220} t_{320}
t_{130} t_{230} t_{330}
v_{10}
v_{20}
v_{30}
v_{10} v_{20} v_{30}
j
i
k

Definition 2.20. FHMDP Compact form
One modification we can make to our matrix form is as follows

$$P^t \in \mathbb{R}^{|\mathcal{A}| \times |\mathcal{S}| \times |\mathcal{S}|}, \quad [P^t]_{a,s,s'} = p(s' \mid s, a) \qquad \text{(State transition tensor)}$$

$$R^t \in \mathbb{R}^{|\mathcal{A}| \times |\mathcal{S}|}, \quad [R^t]_{a,s} = \sum_{s'} p(s' \mid s, a) r(s, a, s') \qquad \text{(Expected reward matrix)}$$

In index form we can then express the Bellman optimality equation as

$$[\mathbf{v}_t]_s = v_*^t(s) = \max_a \left([R^t]_{a,s} + \gamma \sum_{s'} [P^t]_{a,s,s'} [\mathbf{v}_{t+1}]_{s'} \right) \qquad \text{(Bellman optimality (compact index form))}$$

To build up an actual understanding of this form, lets focus on the inner summation term

$$\sum_{s'} [P^t]_{a,s,s'} [\mathbf{v}_{t+1}]_{s'}$$

We can express this in a more expanded form as

$$[P^t_{a_1}, P^t_{a_2}, \ldots, P^t_{a_m}] \times_{s'} \begin{bmatrix} v_*^{t+1}(s_1) \\ v_*^{t+1}(s_2) \\ \vdots \\ v_*^{t+1}(s_n) \end{bmatrix}$$

Each $P^t_{a_i}$ here is a matrix of shape $|\mathcal{S}| \times |\mathcal{S}|$ representing the state transition probabilities for action a_i. So we know that for a given slice a_i, this product yields a vector of shape $|\mathcal{S}|$ expressed as

$$\begin{bmatrix} \sum_{s'} p(s' \mid s_1, a_i) v_*^{t+1}(s') \\ \sum_{s'} p(s' \mid s_2, a_i) v_*^{t+1}(s') \\ \vdots \\ \sum_{s'} p(s' \mid s_n, a_i) v_*^{t+1}(s') \end{bmatrix}$$

The dimensions of the overall product is thus $|\mathcal{A}| \times |\mathcal{S}|$, since we have $|\mathcal{A}|$ such slices.

Definition 2.21. Bellman action-value optimality
The Bellman equation for the optimal action-value function is given by

$$q_*(s, a) = \sum_{(s',r)} p(s', r \mid s, a) \left[r + \gamma \max_{a'} q_*(s', a') \right] \qquad \text{(Bellman action-value optimality)}$$

Its derivation is similar to that of the Bellman optimality equation, but instead of taking the max over actions at the current state, we take it over actions at the next state s'. So we start from the definition of the optimal action-value function and expand the expected return using the configuration probability and the optimal action-value function at the next state. The full derivation is as follows:

$$\begin{aligned} q_*(s, a) &= \mathbb{E}[R_{t+1} + \gamma v_*(S_{t+1}) \mid S_t = s, A_t = a] \\ &= \mathbb{E}\left[R_{t+1} + \gamma \max_{a'} q_*(S_{t+1}, a') \mid S_t = s, A_t = a \right] \\ &= \sum_{(s',r)} p(s', r \mid s, a) \left[r + \gamma \max_{a'} q_*(s', a') \right] \end{aligned}$$