Morris water maze

Reinforcement learning assignment.

Computational Neuroscience

Group RL2: Giorgia Marchesi, Germonda Mooij, Else Meulman

Water maze with place cells - equations

1. Activity i^{th} place cell for current position rat x_t: $f_i(x_t) = exp\left(-\frac{(x_t - s_i)^2}{2\sigma^2}\right)$.
2. Calculate value action cell activities $a_j = \sum_{i=1}^{N} z_{ij} f_i$ for 8 directions j, i sums over N place cells.
3. Actor chooses direction j with probability $P_j = exp(\beta a_j) / \sum_{j=1}^{n_j} exp(\beta a_j)$; $\beta = 2$.
4. Account for momentum: $d\boldsymbol{x}_t = (d\boldsymbol{x}(a_j, v \cdot dt) + m\, d\boldsymbol{x}_{t-1})/(m+1)$, $m = 3$; (dodgy!)

   Calculate new position $\boldsymbol{x}_{t+1} = \boldsymbol{x}_t + d\boldsymbol{x}(a_j, v \cdot dt)$.

   Check whether new position is within pool ($\boldsymbol{x}^2 < 1$), if not reverse by 180 degrees (='bounce'): $d\boldsymbol{x} = -d\boldsymbol{x}$ (this is not an exact reflection but good enough for this problem).

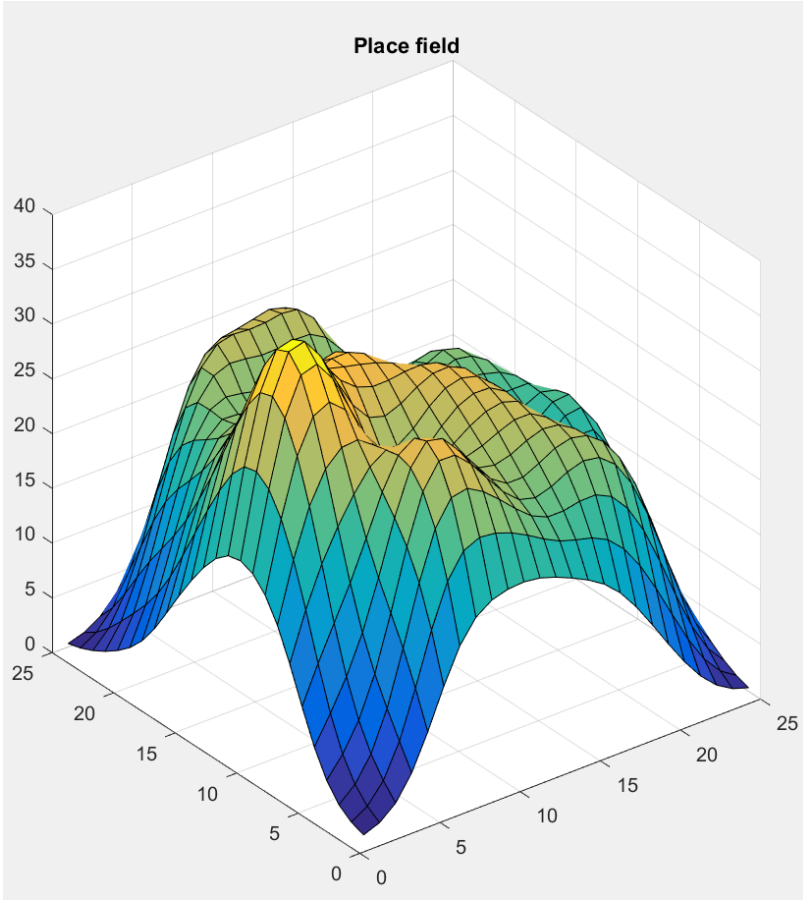
   Recalculate new position $\boldsymbol{x}_{t+1} = \boldsymbol{x}_t + d\boldsymbol{x}(a_j, v \cdot dt)$.

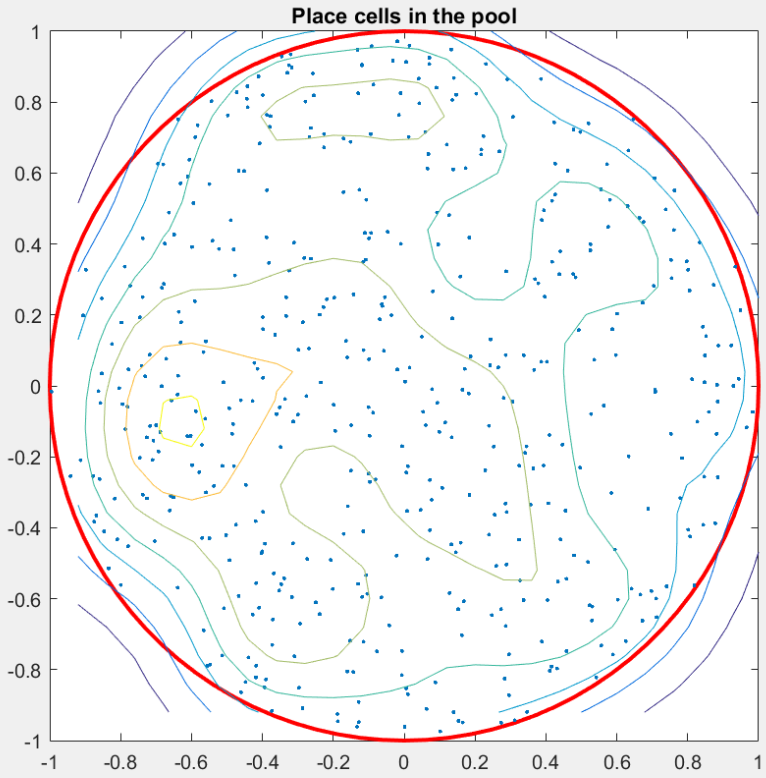
Water maze with place cells - equations

5. Critic evaluates outputs $C_t(x_t) = \sum_{i=1}^{N} w_i f_i(x_t)$ and $C_{t+1}(x_{t+1}) = \sum_{i=1}^{N} w_i f_i(x_{t+1})$.
6. Calculate prediction error $\delta_t = R_{t+1} + \gamma C_{t+1}(x_{t+1}) - C_t(x_t)$.
   $R_{t+1} = 1$ if new position is on platform and 0 otherwise.
7. Critic weights are updated by $\Delta w_i = \varepsilon\, \delta_t f_i(x_t)$; $\varepsilon = 0.1$
8. Actor weights are updated by $\Delta z_{ij} = \varepsilon\, \delta_t f_i(x_t)$ for the selected j; $\varepsilon = 0.1$

Water maze model setup

..

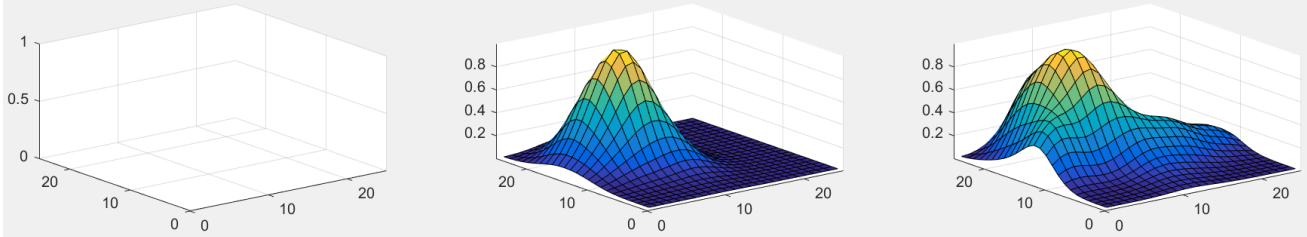




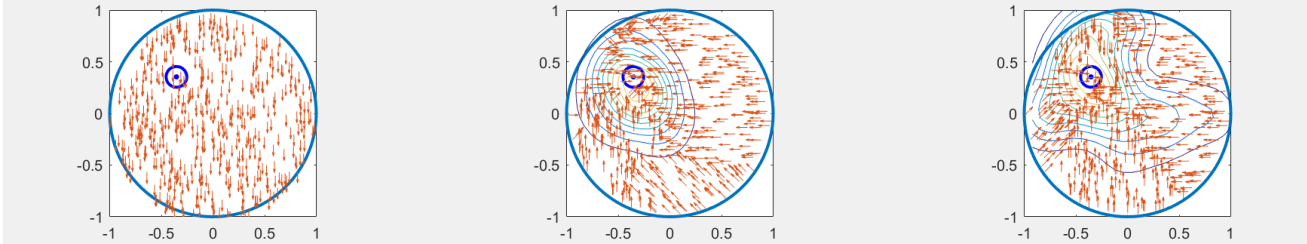
Water maze model results

After 2 trials | After 7 trials | After 22 trials

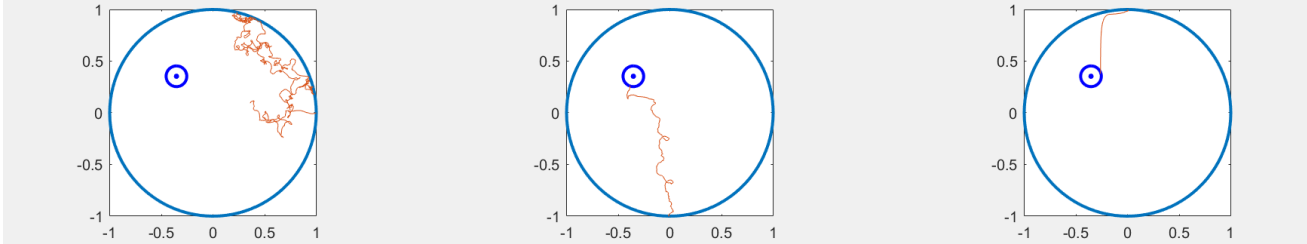
C_p



z_i

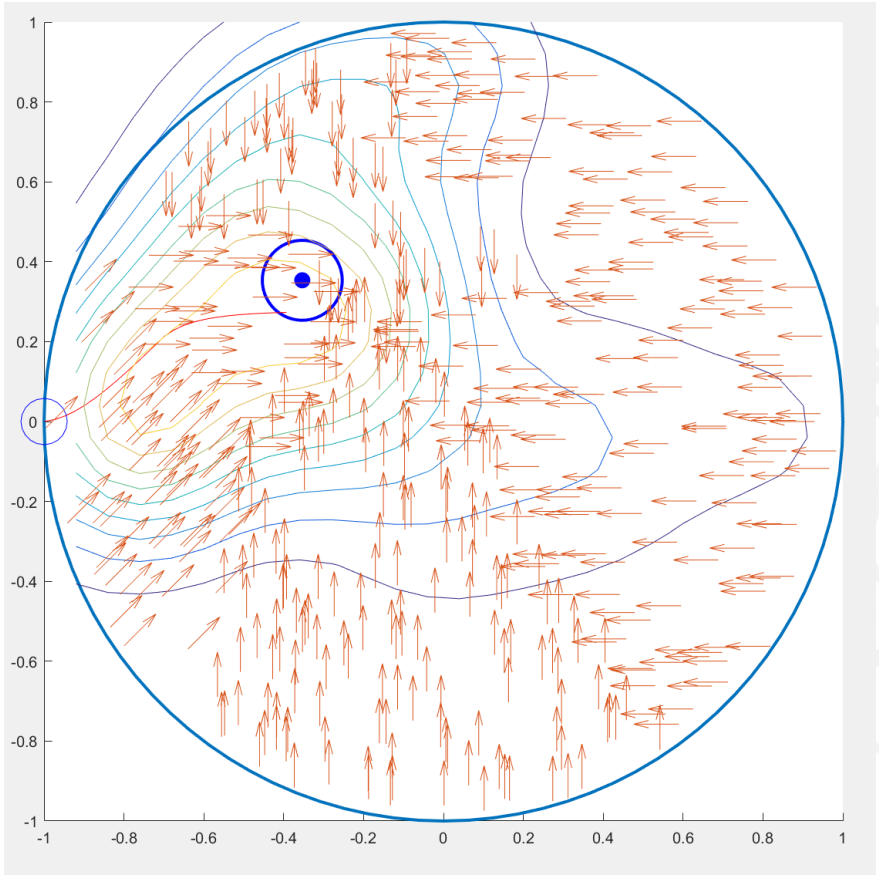


x_t



Water maze model result after 25 trials

..



z_i

Water maze with head direction cells - equations

1. Activity i^{th} place cell for current position rat x_t: $f_i(x_t) = exp\left(-\frac{(x_t - s_i)^2}{2\sigma^2}\right)$.
2. Activity k^{th} head direction cell for current position rat x_t: $g_k(\theta_t) = exp\left(-\frac{(\theta_t - \theta_k)^2}{2\sigma_\theta^2}\right)$.
3. Calculate value action cell activities $a_j = \sum_{i=1}^{N}\sum_{k=1}^{8} z_{ikj} f_i g_k$ for 8 directions j, i sums over N place cells.
4. Actor chooses direction j with probability $P_j = exp(\beta a_j)/\sum_{j=1}^{n_j} exp(\beta a_j)$; $\beta = 2$.
5. Account for momentum: $d\boldsymbol{x}_t = (d\boldsymbol{x}(a_j, v \cdot dt) + m\, d\boldsymbol{x}_{t-1})/(m+1)$, $m = 3$; (now ok!)

   Calculate new position $\boldsymbol{x}_{t+1} = \boldsymbol{x}_t + d\boldsymbol{x}(a_j, v \cdot dt)$.

   Check whether new position is within pool ($\boldsymbol{x}^2 < 1$), if not reverse by 180 degrees (='bounce'): $d\boldsymbol{x} = -d\boldsymbol{x}$ (this is not an exact reflection but good enough for this problem).

   Recalculate new position $\boldsymbol{x}_{t+1} = \boldsymbol{x}_t + d\boldsymbol{x}(a_j, v \cdot dt)$.

Water maze with head direction cells - equations

6. Critic evaluates outputs $C_t(x_t, \theta_t) = \sum_{i=1}^{N} \sum_{k=1}^{8} w_{ik} f_i(x_t) g_k(\theta_t)$ and $C_{t+1}(x_{t+1}, \theta_{t+1}) = \sum_{i=1}^{N} \sum_{k=1}^{8} w_{ik} f_i(x_{t+1}) g_k(\theta_{t+1})$.
7. Calculate prediction error $\delta_t = R_{t+1} + \gamma C_{t+1}(x_{t+1}, \theta_{t+1}) - C_t(x_t, \theta_t)$. $R_{t+1} = 1$ if new position is on platform and 0 otherwise.
8. Critic weights are updated by $\Delta w_{ik} = \varepsilon \delta_t f_i(x_t) g_k(\theta_t)$; $\varepsilon = 0.1$
9. Actor weights are updated by $\Delta z_{ikj} = \varepsilon \delta_t f_i(x_t) g_k(\theta_t)$ for the selected j; $\varepsilon = 0.1$

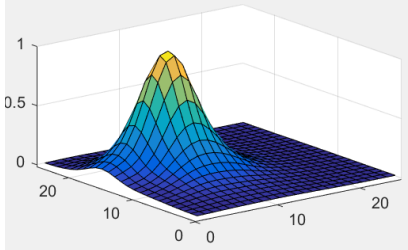
Water maze model results

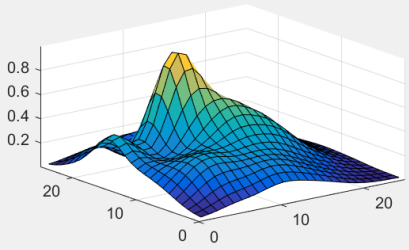
After 20 trials

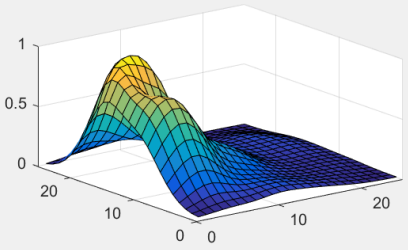
After 70 trials

After 220 trials

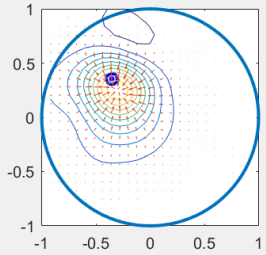
C_p

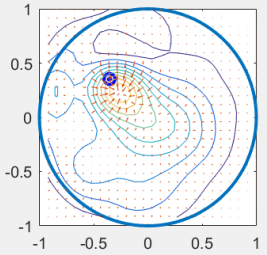


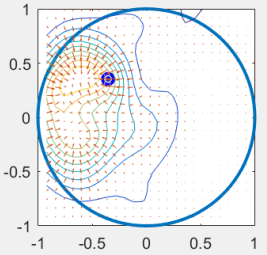




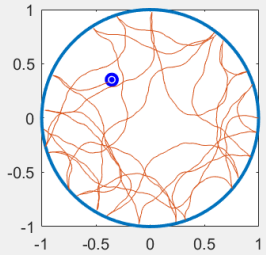
C_p

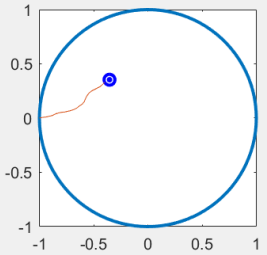


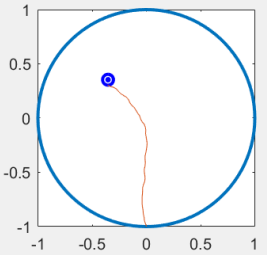




x_t

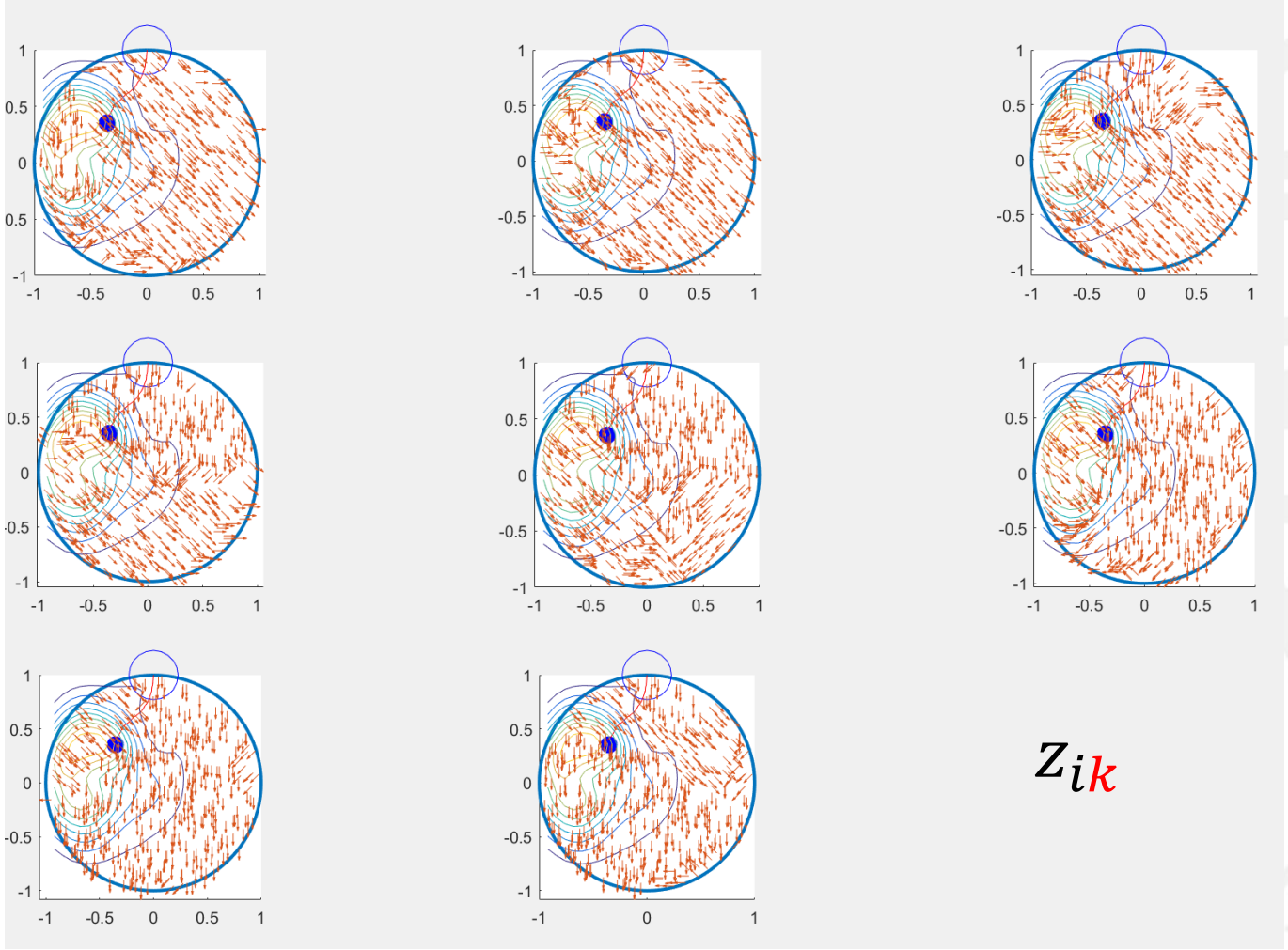






Water maze model result after 25 trials

..



z_{ik}