Learning and Planning with Probabilistic Inference

Machine Learning B

708.062 08W 1sst KU

Dr. Stefan Häusler

Institut für Grundlagen der Informationsverarbeitung

Technische Universität Graz

Course WS 2010/11

Fourth homework set

- **Learning over-hypotheses with probabilistic inference**

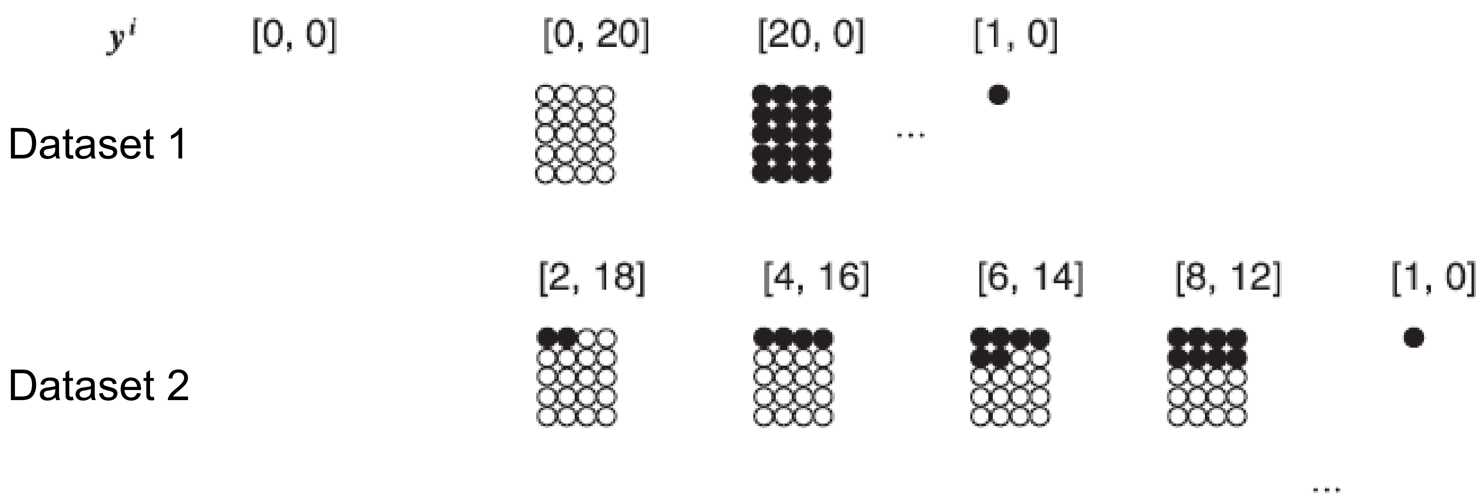
- **Motor planning with probabilistic inference**

Learning with Probabilistic Inference

Bags of Marbles Model:

$N = 20$ bags containing each 20 marbles.

$i = 1,...,N$



Question: What is the color of the remaining marbles in the last bag?

Assignment 11

13 Learning overhypotheses [3 P]

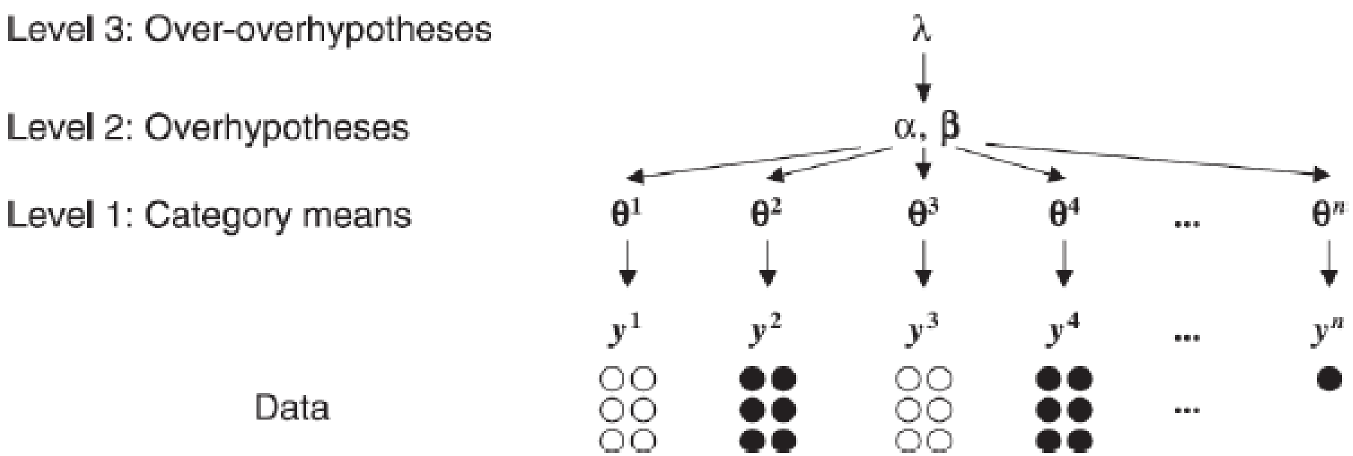
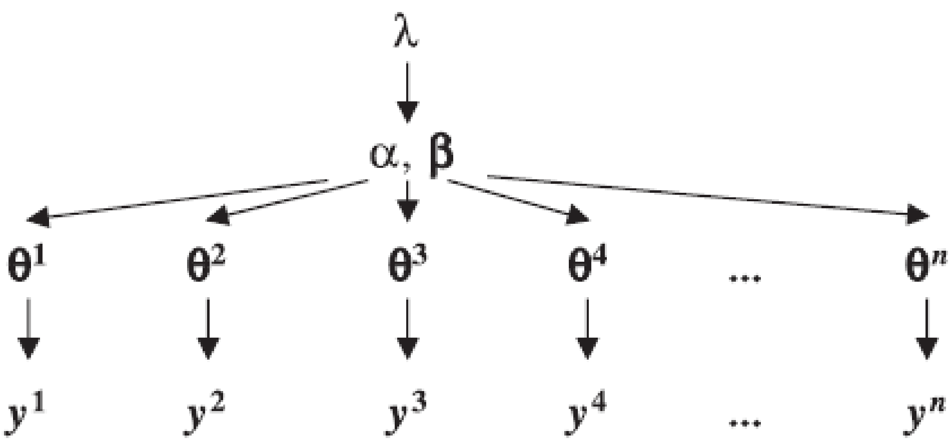


Figure 9: A hierarchical Bayesian model. Each setting of (α, β) is an overhypothesis: β represents the color distribution across all categories, and α represents the variability in color within each category.

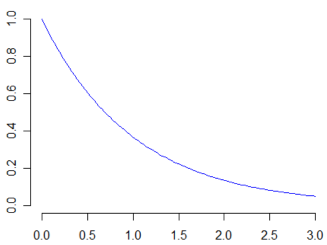
[Kemp et al. 2007, Dev. Sci.]

Hierarchical Bayesian Model



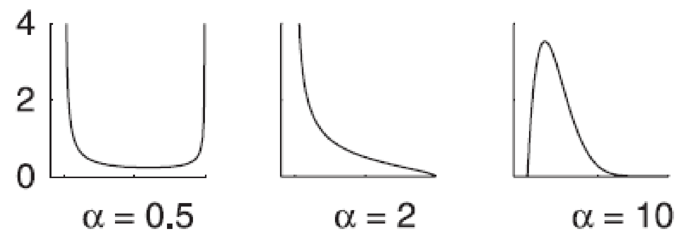
$$
\begin{aligned}
\lambda &= 1 \\
\alpha &\sim Exponential(\lambda) \\
\beta &\sim Beta(1,1) \\
\theta^i &\sim Beta(\alpha,\beta) \\
y^i|n^i &\sim Binomial(\theta^i)
\end{aligned}
$$

$Exponential(\lambda)$



β = [0.2]

$Beta(\alpha,\beta)$



Hierarchical Bayesian Model

$$P(\mathbf{y}^1, ..., \mathbf{y}^n, \theta^1, ..., \theta^n, \alpha, \beta|\lambda) = \prod_{i=1}^{n} P(\mathbf{y}^i|\theta^i)P(\theta^i|\alpha, \beta)P(\alpha|\lambda)P(\beta) \quad (1)$$

with

$$\begin{aligned} \alpha &\sim Exponential(\lambda) && (2)\\ \beta &\sim Beta(1,1) && (3)\\ \theta^i &\sim Beta(\alpha,\beta) && (4)\\ \mathbf{y}^i|n^i &\sim Binomial(\theta^i) && (5) \end{aligned}$$

where n^i is the number of observations for bag i.

Gibbs Sampling

Consider the distribution $p(\boldsymbol{x}) = p(x_1, \ldots, x_M)$

Gibbs Sampling

1. Initialize $\{x_i : i = 1, \ldots, M\}$
2. For $\tau = 1, \ldots, T$:
   - Sample $x_i^{(\tau+1)} \sim p(x_i | x_2^{(\tau)}, x_3^{(\tau)}, \ldots, x_M^{(\tau)})$.

i ... is chosen randomly from {1,... ,M}

Resampling from the Color Distribution

Resample θ^i

$$P(\mathbf{y}^i|\theta^i)P(\theta^i|\alpha,\beta)$$

θ^i are vectors: If 60% of the marbles in bag 7 are black, then θ^7 = [0.6, 0.4].

$\mathbf{y}^i$ are fixed observations: If all but one marbles in bag 7 are black $\mathbf{y}^7$ = [4, 1].

(α,β) are fixed parameters of all color distribution.

Resample from: $$P(\theta_1^i) = Beta(\alpha + y_1^i, \beta + y_2^i)$$

MATLAB: `random('beta',α,β)`

Resampling the Overhypotheses

Resample (α, β) (but each parameter indepentently)

$$P(\theta^i|\alpha,\beta)P(\alpha|\lambda)P(\beta)$$

$$\begin{aligned} \alpha &\sim Exponential(\lambda) \\ \beta &\sim Beta(1,1) \\ \theta^i &\sim Beta(\alpha,\beta) \end{aligned}$$

Prob. distribution has not a simple form.

2 Solutions: a) Discretize the space of α and β

b) Perform sampling-importance-resampling

Sampling-importance-resampling

The sampling-importance-resampling (SIR) approach makes use of a sampling distribution $q(z)$. (therefore z is either α or β)

There are three stages to the scheme:

1. L samples $z^{(1)}, \ldots, z^{(L)}$ are drawn from $q(z)$.

2. Weights $w_1, \ldots, w_L$ are constructed using $p(z) \sim P(\theta^i|\alpha,\beta)P(\alpha|\lambda)P(\beta)$

$$w_l = \frac{\widetilde{r}_l}{\sum_m \widetilde{r}_m} \qquad r_l = p(\mathbf{z}^{(l)})/q(\mathbf{z}^{(l)}) \qquad \text{where } l, m = 1,\ldots,L$$

3. Finally, a sample is drawn from the discrete distribution $(z^{(1)}, \ldots, z^{(L)})$ with probabilities given by the weights $(w_1, \ldots, w_L)$.

Guarantee: For infinitely large L $p(z)$ approaches $q(z)$.

Tasks

1. After observing 10 all-white bags, 10 all-black bag and a single black marble in the last bag.
2. After observing 20 mixed bags, where half of the marbles are white and half of the marbles are black, and a single black marble in the last bag.
3. Same as in 1 but with fixed $\alpha = 1$ and $\beta = 0.5$.
4. Same as in 2 but with fixed $\alpha = 1$ and $\beta = 0.5$.

Calculate average distributions across 50 Markov chains, each of which was run for 100000 iterations (discard the first 10000 samples as burn-in).

Hand in plots for all estimated distributions and interpret your results.

Hints

Hints:

1. For the resampling of θ^i the factor $P(\mathrm{y}^i|\theta^i)P(\theta^i|\alpha,\beta)$ is again a Beta distribution that can be directly sampled in MATLAB with the command `random` and the argument `beta`.

2. For the resampling of α and β apply sampling-importance-resampling sampling (with 100 samples drawn from a uniform proposal distribution) from a distribution proportional to the factor $P(\theta^i|\alpha,\beta)P(\alpha|\lambda)P(\beta)$.

3. You should adapt your own MATLAB code from the previous Gibbs sampling homework example to solve this assignment.

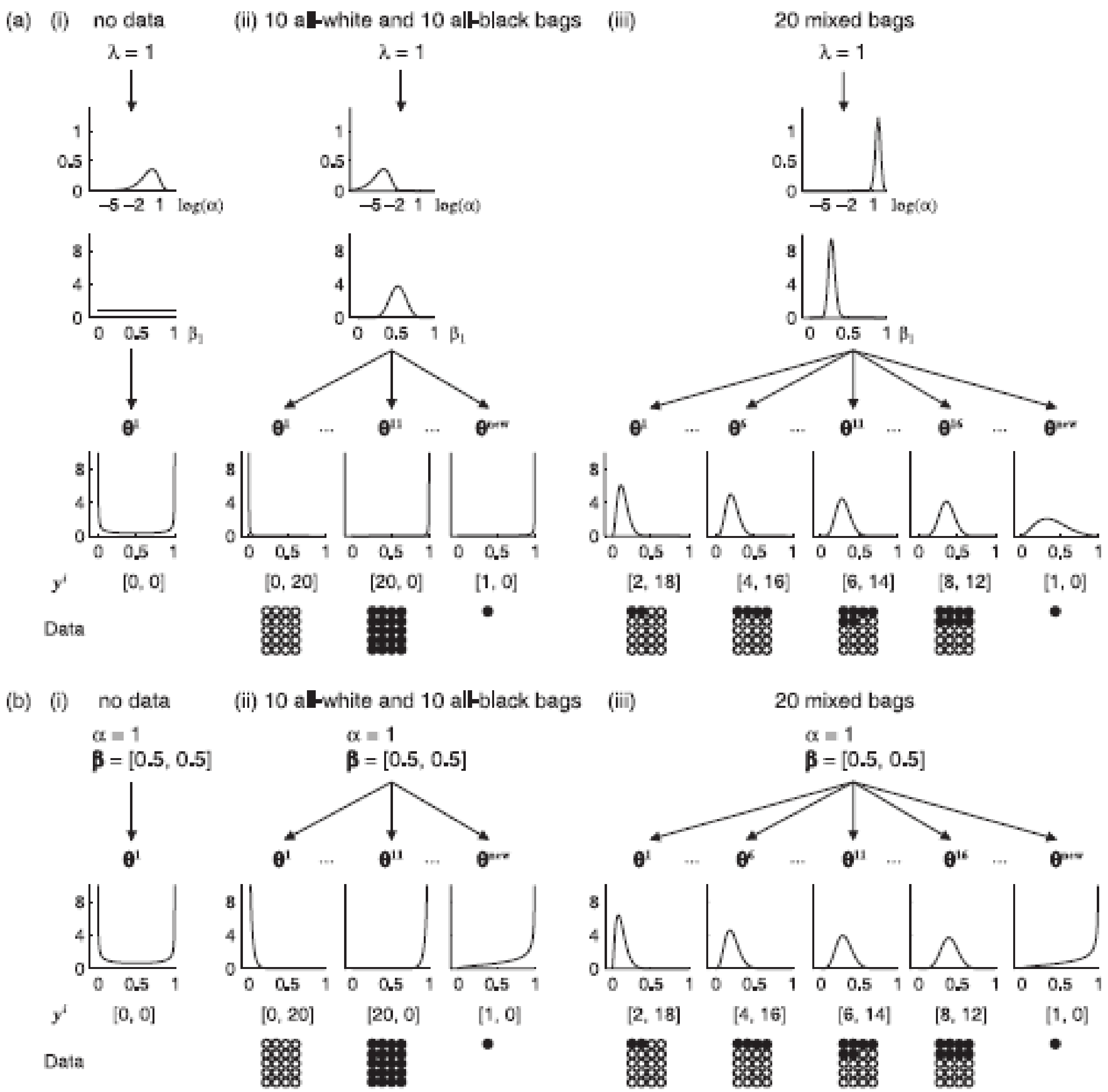
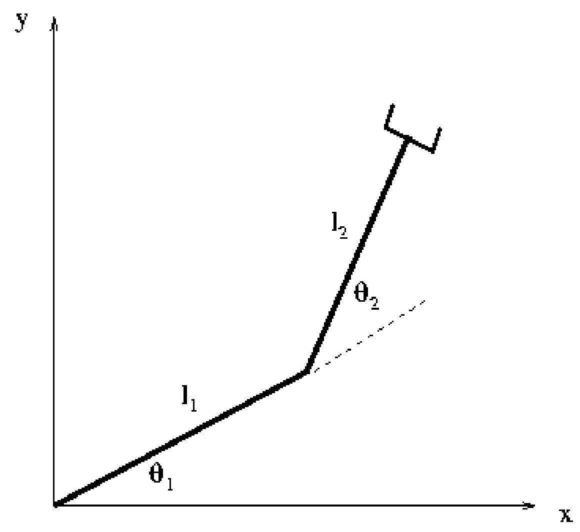


Fig 3 in [Kemp et al. 2007, Dev. Sci.]

Planning with Probabilistic Inference

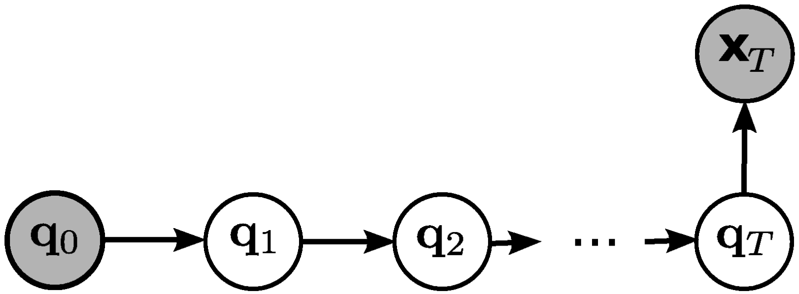
In this task one has to plan an optimal path from an initial joint-position to a given endeffector-position with Gibbs sampling.



Dynamic Bayesian Network

Goal:

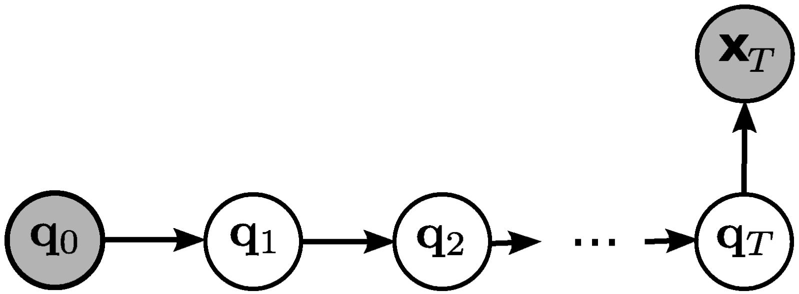
We want to reach our target within T = 10 time steps,
i.e. we get a dynamic Bayesian Network with one node per time step (11 nodes),



Each node t represents the joint positions $\boldsymbol{q}_t$ at time t. For simplicity, we willuse a discrete representation of the joint positions.

Prior distributions

We use Gaussian motion prior in order to define the transition probabilities



Let $\mathbf{q}^{(j)}$ be the joint position vector (in radiants), then

$$P(\mathbf{q}_t^{(j)} | \mathbf{q}_{t-1}^{(i)}) \propto \mathcal{N}(\mathbf{q}_t^{(j)} | \mathbf{q}_{t-1}^{(i)}, \mathrm{W}) \qquad \mathbf{q}_0 \text{ is } [\pi/4, 0]^T$$

where W equals diag([0.0125, 0.05]).

The motion prior encodes our laziness, meaning we do not want to move.

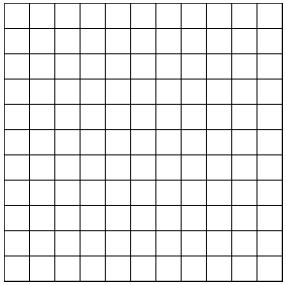
Kinematic task space mapping

We will also use a discrete representation for task space (Cartesian coordinates of the hand).

We use again a 11×11 uniform grid over the range [0; 1] for x and y.

The probability of reaching the jth discrete task space position when being in the ith discrete joint space position is given by

$$P(x_t^{(j)} | q_t^{(i)}) \propto \mathcal{N}(x_t^{(j)} | \Phi(q_t^{(i)}), \mathrm{C})$$



Φ … non-linear mapping from the joint positions to the endeffector coordinates

C … covariance matrix set to diag([0.004, 0.004]).

Task

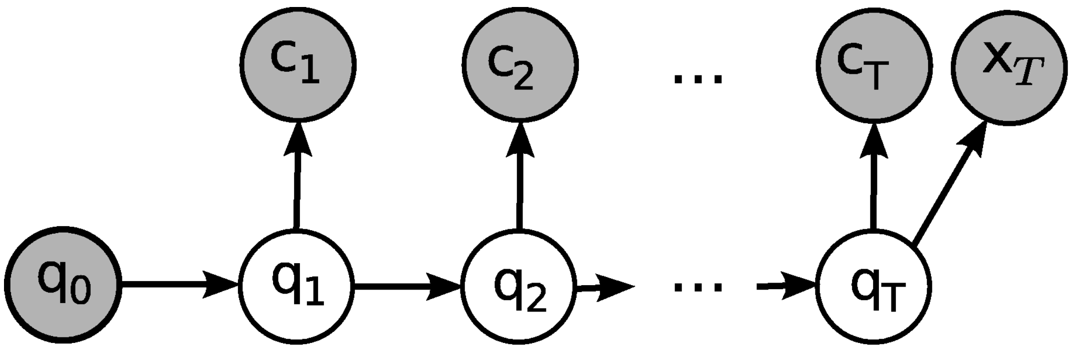
- Your main task is to generate the state transition probabilities P_{qq} as well as the kinematic task mapping P_{qx} as described in the text.

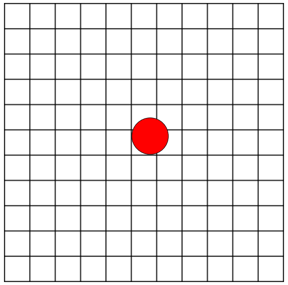
- Initialize your trajectory $\mathbf{q}_{1:T}$ with random discrete indices. Use 5000 Gibbs sampling steps to estimate the true distribution. Plot 5 independent samples of $\mathbf{q}_{1:T}$ from the sampling process. We assume that two samples are independent at least after 100 Gibbs Sampling steps.

- Now we want to calculate the marginals $P(\mathbf{q}_t = q^{(i)})$ for all i and t. Therefore count the number of of times in which the variable $\mathbf{q}_t$ equals $q^{(i)}$ during the last 3000 Gibbs sampling steps. Plot the marginals for each time step, use a visualization which you find appropriate. How does the estimated solution look like?

- Repeat the experiment at least 10 times using different initial positions for the Gibbs sampler. Are the marginals always similar (up to a certain noise level...)? If not, why not?

Planning with Obstacle Avoidance





For simplicity, we assume that only the endeffector can collide with the obstacle with collision probability

$$P(c_t = 1 | q_t^{(j)}) \;=\; \exp(-1/2 ||\Phi(\mathbf{q}^{(j)}) - [0.5, 0.5]^T||^2 / 0.15^2)$$

For obstacle avoidance we set $c_t = 0$.

Task

- Generate the collision probability P_{qc}
- Use Gibbs sampling the same ways as before, visualize the marginals $P(\mathbf{q}_t = q^{(i)})$. How has the estimated solution changed?

Pseudo-Dynamic Planning

Now we also want to add the velocities $\dot{q}$ of the joints to our planning scenario. Therefore, we will also incorporate controls u of the robot in our model.

$$P(\mathbf{q}_t, \dot{\mathbf{q}}_t | \mathbf{q}_{t-1}, \dot{\mathbf{q}}_{t-1}, \mathbf{u}_{t-1}) = \mathcal{N}([\mathbf{q}_t; \dot{\mathbf{q}}_t] | [\mathbf{q}_{t-1} + 0.1\dot{\mathbf{q}}_{t-1}; \dot{\mathbf{q}}_{t-1} + 0.1\mathbf{u}_{t-1}], \mathbf{W})$$

where $\mathbf{W}$ is set to $\mathrm{diag}([10^{-5}, 10^{-5}, 10^{-3}, 10^{-3}])$.

The actions are unknown variables that are integrated out:

$$P(\mathbf{q}_t, \dot{\mathbf{q}}_t | \mathbf{q}_{t-1}, \dot{\mathbf{q}}_{t-1}) = \sum_{i=1}^{25} P(\mathbf{q}_t, \dot{\mathbf{q}}_t | \mathbf{q}_{t-1}, \dot{\mathbf{q}}_{t-1}, \mathbf{u}_{t-1}^{(i)}) P(\mathbf{u}_{t-1}^{(i)})$$

The action priors : $P(\mathbf{u}_{t-1}^{(i)}) = \mathcal{N}(\mathbf{u}_{t-1}^{(i)} | 0, \mathbf{H})$, where $\mathbf{H}$ is set to $16\mathbf{I}$.