1 HMMs: Warmup

1. What are the three components of a hidden markov model? What makes it "hidden"?

   - Initial distribution: $P(X_0)$
   - Transition model: $P(X_t|X_{t-1})$
   - Sensor model: $P(E_t|X_t)$

   The hidden part of hidden markov models comes from the fact that we do not observe the state variables X_i directly, rather we observe the evidence variables E_i and must make conclusions about the underlying true state.

2. Write an expression for the joint distribution of a hidden markov model consisting of states $X_0, \ldots, X_n$ and evidence variables $E_1, \ldots, E_N$. How does the expression reflect the underlying structure of the model?

   $P(X_0, \ldots, X_N, E_1, \ldots, E_N) = P(X_0) \prod_{t=1}^{N} P(X_t|X_{t-1})P(E_t|X_t)$

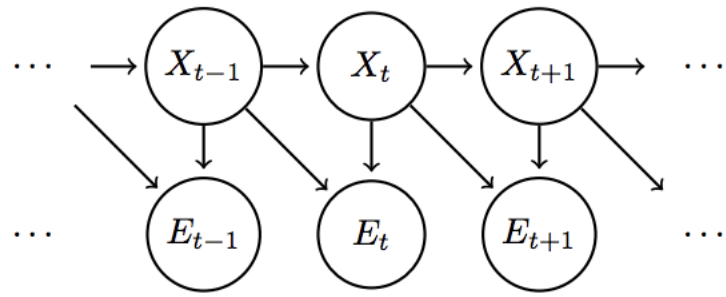
   This expression reflects that the a state is only directly influenced by its previous state, and that the evidence is independent of everything else given the corresponding state.

3. For each of the following descriptions in English of an inference task, write the corresponding probability expression:

   - Draw conclusions about our current underlying state given evidence up to the current time step
   - Draw conclusions about our future underlying state given evidence up to the current time step
   - Draw conclusions about a past underlying state given evidence up to the current time step
   - Draw conclusions about the sequence of underlying states given evidence up to the current time step
   - Draw conclusions about the most likely sequence of underlying states given evidence up to the current time step

   - Filtering: $P(X_t|E_{1:t})$
   - Prediction: $P(X_{t+k}|E_{1:t}), k > 0$
   - Smoothing: $P(X_k|E_{1:t}), 1 \leq k < t$
   - Explanation: $P(X_{1:t}|E_{1:t})$
   - Most likely explanation: $\text{argmax}_{X_{1:t}} P(X_{1:t}|E_{1:t})$

4. Hidden Markov Models can be extended in a number of ways to incorporate additional relations. Since the independence assumptions are different in these extended Hidden Markov Models, the forward algorithm updates will also be different. What is the forward algorithm updates for the extended Hidden Markov Models specified by the following Bayes net?



$P(X_t|e_{1:t}) = \alpha \sum_{x_{t-1}} P(e_t|x_t, x_{t-1}) P(x_t|x_{t-1}) P(x_{t-1}|e_{1:t-1})$

2 HMMs: Tracking a Jabberwock

You have been put in charge of a Jabberwock for your friend Lewis. The Jabberwock is kept in a large tugley wood which is conveniently divided into an $N \times N$ grid. It wanders freely around the N^2 possible cells. At each time step $t = 1, 2, 3, \ldots$, the Jabberwock is in some cell $X_t \in \{1, \ldots, N\}^2$, and it moves to cell X_{t+1} randomly as follows: with probability $1 - \epsilon$, it chooses one of the (up to 4) valid neighboring cells uniformly at random; with probability ϵ, it uses its magical powers to teleport to a random cell uniformly at random among the N^2 possibilities (it might teleport to the same cell). Suppose $\epsilon = \frac{1}{2}$, $N = 10$ and that the Jabberwock always starts in $X_1 = (1, 1)$.

(a) Compute the probability that the Jabberwock will be in $X_2 = (2, 1)$ at time step 2. What about $P(X_2 = (4, 4))$?

$P(X_2 = (2, 1)) = 1/2 \cdot 1/2 + 1/2 \cdot 1/100 = 0.255$
$P(X_2 = (4, 4)) = 1/2 \cdot 1/100 = 0.005$

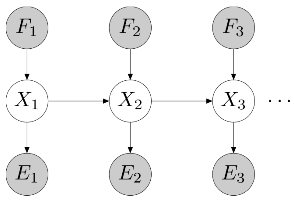
At each time step t, you don't see X_t but see E_t, which is the row that the Jabberwock is in; that is, if $X_t = (r, c)$, then $E_t = r$. You still know that $X_1 = (1, 1)$.

(b) Suppose we see that $E_1 = 1$, $E_2 = 2$. Fill in the following table with the distribution over X_t after each time step, taking into consideration the evidence. Your answer should be concise. <u>Hint</u>: you should not need to do any heavy calculations.

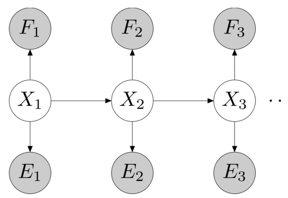
<table>
<tr><th>t</th><th>$P(X_t \mid e_{1:t-1}, X_1 = (1,1))$</th><th>$P(X_t \mid e_{1:t}, X_1 = (1,1))$</th></tr>
<tr><td>1</td><td><table><tr><th>X_1</th><th>$P(X_1)$</th></tr><tr><td>(1, 1)</td><td></td></tr><tr><td>all other values</td><td></td></tr></table></td><td><table><tr><th>X_1</th><th>$P(X_1)$</th></tr><tr><td>(1, 1)</td><td></td></tr><tr><td>all other values</td><td></td></tr></table></td></tr>
<tr><td>2</td><td><table><tr><th>X_2</th><th>$P(X_2 \mid e_1, X_1 = (1,1))$</th></tr><tr><td>(1, 2)</td><td></td></tr><tr><td>(2, 1)</td><td></td></tr><tr><td>all other values</td><td></td></tr></table></td><td><table><tr><th>X_2</th><th>$P(X_2 \mid e_{1:2}, X_1 = (1,1))$</th></tr><tr><td>(2, 1)</td><td></td></tr><tr><td>$(2, a)$ $(\forall a, a > 1)$</td><td></td></tr><tr><td>all other values</td><td></td></tr></table></td></tr>
</table>

<table>
<tr><th>t</th><th>$P(X_t \mid e_{1:t-1}, X_1 = (1,1))$</th><th>$P(X_t \mid e_{1:t}, X_1 = (1,1))$</th></tr>
<tr><td>1</td><td><table><tr><th>X_1</th><th>$P(X_1)$</th></tr><tr><td>(1, 1)</td><td>1</td></tr><tr><td>all other values</td><td>0</td></tr></table></td><td><table><tr><th>X_1</th><th>$P(X_1)$</th></tr><tr><td>(1, 1)</td><td>1</td></tr><tr><td>all other values</td><td>0</td></tr></table></td></tr>
<tr><td>2</td><td><table><tr><th>X_2</th><th>$P(X_2 \mid e_1, X_1 = (1,1))$</th></tr><tr><td>(1, 2)</td><td>51/200</td></tr><tr><td>(2, 1)</td><td>51/200</td></tr><tr><td>all other values</td><td>1/200</td></tr></table></td><td><table><tr><th>X_2</th><th>$P(X_2 \mid e_{1:2}, X_1 = (1,1))$</th></tr><tr><td>(2, 1)</td><td>51/60</td></tr><tr><td>$(2, a)$ $(\forall a, a > 1)$</td><td>1/60</td></tr><tr><td>all other values</td><td>0</td></tr></table></td></tr>
</table>

You are a bit unsatisfied that you can't pinpoint the Jabberwock exactly. But then you remembered Lewis told you that the Jabberwock teleports only because it is frumious on that time step, and it becomes frumious independently of anything else. Let us introduce a variable $F_t \in \{0, 1\}$ to denote whether it will teleport at time t. We want to to add these frumious variables to the HMM. Consider the two candidates:



(A)



(B)

(A)	(B)
$X_1 \perp\!\!\!\perp X_3 \mid X_2$	$X_1 \perp\!\!\!\perp X_3 \mid X_2$
$X_1 \perp\!\!\!\perp E_2 \mid X_2$	$X_1 \perp\!\!\!\perp E_2 \mid X_2$
$X_1 \perp\!\!\!\perp F_2 \mid X_2$	$X_1 \perp\!\!\!\perp F_2 \mid X_2$
$X_1 \perp\!\!\!\perp E_4 \mid X_2$	$X_1 \perp\!\!\!\perp E_4 \mid X_2$
$X_1 \perp\!\!\!\perp F_4 \mid X_2$	$X_1 \perp\!\!\!\perp F_4 \mid X_2$
$E_3 \perp\!\!\!\perp F_3 \mid X_3$	$E_3 \perp\!\!\!\perp F_3 \mid X_3$
$E_1 \perp\!\!\!\perp F_2 \mid X_2$	$E_1 \perp\!\!\!\perp F_2 \mid X_2$
$E_1 \perp\!\!\!\perp F_2 \mid E_2$	$E_1 \perp\!\!\!\perp F_2 \mid E_2$

(c) For each model, circle the conditional independence assumptions above which are true in that model.

(A)	(B)
$X_1 \perp\!\!\!\perp X_3 \mid X_2$ ✓	$X_1 \perp\!\!\!\perp X_3 \mid X_2$ ✓
$X_1 \perp\!\!\!\perp E_2 \mid X_2$ ✓	$X_1 \perp\!\!\!\perp E_2 \mid X_2$ ✓
$X_1 \perp\!\!\!\perp F_2 \mid X_2$	$X_1 \perp\!\!\!\perp F_2 \mid X_2$ ✓
$X_1 \perp\!\!\!\perp E_4 \mid X_2$ ✓	$X_1 \perp\!\!\!\perp E_4 \mid X_2$ ✓
$X_1 \perp\!\!\!\perp F_4 \mid X_2$ ✓	$X_1 \perp\!\!\!\perp F_4 \mid X_2$ ✓
$E_3 \perp\!\!\!\perp F_3 \mid X_3$ ✓	$E_3 \perp\!\!\!\perp F_3 \mid X_3$ ✓
$E_1 \perp\!\!\!\perp F_2 \mid X_2$	$E_1 \perp\!\!\!\perp F_2 \mid X_2$ ✓
$E_1 \perp\!\!\!\perp F_2 \mid E_2$	$E_1 \perp\!\!\!\perp F_2 \mid E_2$

(d) Which Bayes net is more appropriate for the problem domain here, (A) or (B)? Justify your answer.

(A) because the choice of X depends on F in the problem description.

For the following questions, your answers should be fully general for models of the structure shown above, not specific to the teleporting Jabberwock.

(e) For (A), express $P(X_{t+1}, e_{1:t+1}, f_{1:t+1})$ in terms of $P(X_t, e_{1:t}, f_{1:t})$ and the conditional probability tables used to define the network. Assume the E and F nodes are all observed.

$$P(x_{t+1}, e_{1:t+1}, f_{1:t+1}) = P(e_{t+1}|x_{t+1})P(f_{t+1}) \sum_{x_t} P(x_{t+1}|x_t, f_{t+1})P(x_t, e_{1:t}, f_{1:t}).$$

We're already provided with $P(x_t, e_{1:t}, f_{1:t})$. To get $P(x_t + 1, e_{1:t}, f_{1:t})$, we can sum over all x_t and multiply by $P(x_{t+1} \mid x_t, f_{t+1})$, the conditional probability table of x_{t+1}.
Then, to get the joint probability $P(x_t + 1, e_{1:t+1}, f_{1:t+1})$, we multiply the above quantity with the emission probability ($P(e_{t+1} \mid x_{t+1})$) and $P(f_{t+1})$, the CPT of $P(f_{t+1})$.

(f) For (B), express $P(X_{t+1}, e_{1:t+1}, f_{1:t+1})$ in terms of $P(X_t, e_{1:t}, f_{1:t})$ and the CPTs used to define the network. Assume the E and F nodes are all observed.

$$P(x_{t+1}, e_{1:t+1}, f_{1:t+1}) = P(e_{t+1}|x_{t+1})P(f_{t+1}|x_{t+1}) \sum_{x_t} P(x_{t+1}|x_t)P(x_t, e_{1:t}, f_{1:t}).$$

Similar idea as above, except this time we multiply the joint probability by $P(x_{t+1}|x_t)$, since x_{t+1} now no longer depends on f_{t+1}).

Suppose that we don't actually observe the F_ts.

(g) For (A), express $P(X_{t+1}, e_{1:t+1})$ in terms of $P(X_t, e_{1:t})$ and the CPTs used to define the network.

$$P(x_{t+1}, e_{1:t+1}) = P(e_{t+1}|x_{t+1}) \sum_{f_{t+1}} P(f_{t+1}) \sum_{x_t} P(x_{t+1}|x_t, f_{t+1})P(x_t, e_{1:t}).$$

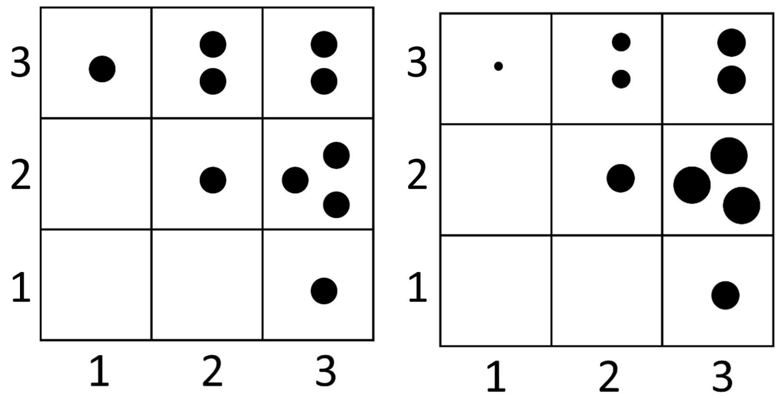
(h) For (B), express $P(X_{t+1}, e_{1:t+1})$ in terms of $P(X_t, e_{1:t})$ and the CPTs used to define the network.

$$P(x_{t+1}, e_{1:t+1}) = P(e_{t+1}|x_{t+1}) \sum_{x_t} P(x_{t+1}|x_t)P(x_t, e_{1:t}).$$

For (g) and (h), we essentially use the same logic as (e) and (f). However, we no longer need the F_ts in the joint probability - so for any probability values that are conditioned on an f_t, we multiply by $P(f_t)$ and sum over all possible f_t values. If not (i.e., for graph (B)), we simply drop that term when computing the joint probability.

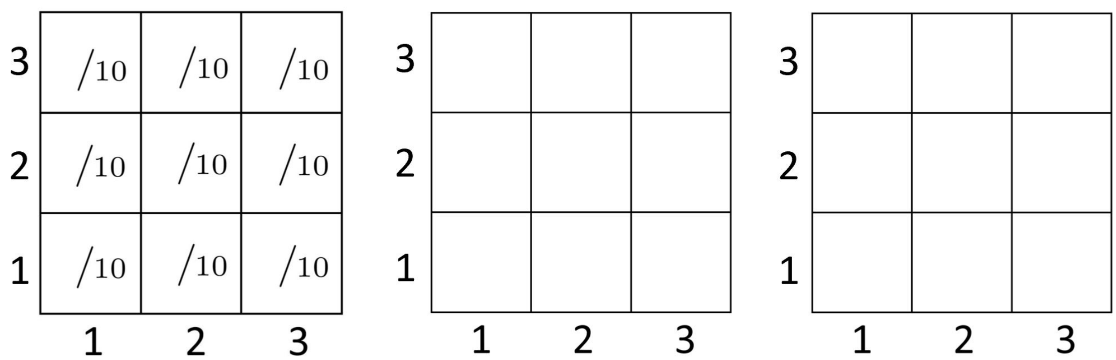
3 Particle Filtering: Warmup

(a) The following state space contains 10 particles. The left grid shows the prior belief distribution of the particles at time t, while the grid on the right shows the states weighted by the observations $P(e_t|S_t)$.



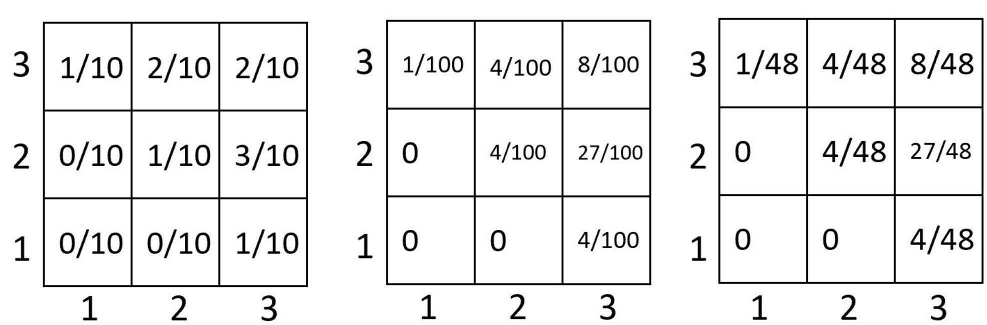
State	Weight
(1,3)	0.1
(2,2)	0.4
(2,3)	0.2
(3,1)	0.4
(3,2)	0.9
(3,3)	0.4

Fill in the following grids to update the belief distribution. Each square in the "Belief" grid should correspond to $\hat{P}(S_t|e_{1:t-1})$, the estimated probability of a particle being in state S at time t. Each square in the "Unnormalized" grid should correspond to the probability $P(S_t, e_t|e_{1:t-1})$. The "Normalized" grid should contain our updated belief distribution $\hat{P}(S_t|e_t, e_{1:t-1})$.



Belief Unnormalized Normalized

Solution: Note that states which did not appear in the weight table have a weight of 0.



Belief Unnormalized Normalized

(b) True / False: The particle filtering algorithm is consistent since it gives correct probabilities as the number of samples N tends to infinity.

True

(c) True / False: The number of samples we use in the particle filtering algorithm increases from one time step to the next.

False. The number of samples stays constant from one time step to the next. The last step for each iteration of the algorithm is resampling, which builds a new population of N samples from the belief distribution updated by observation weights.

4 Tracking the Jabberwock

Lewis' Jabberwock is in the wild: its position is in a two-dimensional discrete grid, but this time the grid is not bounded. In other words, the position of the Jabberwock is a pair of integers $z = (x, y) \in \mathbb{Z}^2 = \{\dots, -2, -1, 0, 1, 2, \dots\} \times \{\cdots, -2, -1, 0, 1, 2, \cdots\}$. At each time step $t = 1, 2, 3, \dots$, the Jabberwock is in some cell $Z_t = z \in \mathbb{Z}^2$, and it moves to cell Z_{t+1} randomly as follows: with probability 1/2, it stays where it is; otherwise, it chooses one of its four neighboring cells uniformly at random (fortunately, no teleportation is allowed this week!).

(a) Write a function for the transition probability $P(Z_{t+1} = (x', y') | Z_t = (x, y))$.

$$P(Z_{t+1} = (x', y') | Z_t = (x, y)) = \begin{cases} \frac{1}{2} & \text{if } x = x', y = y' \\ \frac{1}{8} & \text{if } |x - x'| + |y - y'| = 1 \\ 0 & \text{otherwise} \end{cases}$$

We will use the particle filtering algorithm to track the Jabberwock. As a source of randomness use values in order from the following sequence $\{a_i\}_{1 \le i \le 14}$. Use these values to sample from any discrete distribution of the form $P(X)$ where X takes values in $\{1, 2, \dots, N\}$. Given $a_i \sim U[0, 1]$, return j such that $\sum_{k=1}^{j-1} P(X = k) \le a_i < \sum_{k=1}^{j} P(X = k)$. You have to fix an ordering of the elements for this procedure to make sense.

a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	a_9	a_{10}	a_{11}	a_{12}
0.142	0.522	0.916	0.792	0.703	0.231	0.036	0.859	0.677	0.221	0.156	0.249

At each time step t you get an observation of the x coordinate R_t in which the Jabberwock sits, but it is a noisy observation. Given the true position $Z_t = (x, y)$, you observe the correct value according to the following probability:

$$P(R_t = r | Z_t = (x, y)) \propto (0.5)^{|x-r|}$$

(b) Suppose that you know that half of the time, the Jabberwock starts at $z_1 = (0, 0)$, and the other half, at $z_1 = (1, 1)$. You get the following observations: $R_1 = 1, R_2 = 0, R_3 = 1$. Fill out the table for each time step using a particle filter with 2 particles to compute an approximation to $P(Z_1, Z_2, Z_3 | r_1, r_2, r_3)$. Sample transitions from the table below using the a_i's as our source of randomness. The a_i's you should use for each step habe been indicated in the last row of each table. Note that going "left" decrements the x-coordinate by 1, and going "down" decrements the y-coordinate by 1.

[0; 0.5)	Stay
[0.5; 0.625)	Up
[0.625; 0.75)	Left
[0.75; 0.875)	Right
[0.875; 1)	Down

Initial	Belief $\hat{P}(z_1)$	Weights $P(r_1\|z_1)$	Unnormalized $\hat{P}(z_1, r_1)$	Normalized $\hat{P}(z_1\|r_1)$	Resampling
$p_1 =$ (0,0) $p_2 =$ (1,1) a_1, a_2	1/2 1/2				$p_1 =$ (,) $p_2 =$ (,) a_3, a_4

Transition $P(z_2\|z_1)$	Belief $\hat{P}(z_2\|r_1)$	Weights $P(r_2\|z_2)$	Unnormalized $\hat{P}(z_2, r_2\|r_1)$	Normalized $\hat{P}(z_2\|r_1, r_2)$	Resampling
$p_2 =$ (,) $p_2 =$ (,) a_5, a_6					$p_1 =$ (,) $p_2 =$ (,) a_7, a_8

Transition $P(z_3\|z_2)$	Belief $\hat{P}(z_3\|r_1, r_2)$	Weights $P(r_3\|z_3)$	Unnormalized $\hat{P}(z_3, r_3\|r_1, r_2)$	Normalized $\hat{P}(z_3\|r_1, r_2, r_3)$	Resampling
$p_1 =$ (,) $p_2 =$ (,) a_9, a_{10}					$p_1 =$ (,) $p_2 =$ (,) a_{11}, a_{12}

For each time step, we use our random numbers a_i to sample from the prior or from the transitions. Next, we find the weight of the sample based on the observation at that time step. We update our belief distribution with the weight by taking the product $\hat{P}(z_t|r_{1:t-1})P(r_t|z_t)$ and normalizing to get $\hat{P}(z_t|r_{1:t})$. Note that since the two particles are in different locations at each time step, the belief $\hat{P}(z_t|r_{1:t-1})$ is always 1/2. Finally, we resample the particles from this updated belief distribution.

Initial	Belief $\hat{P}(z_1)$	Weights $P(r_1\|z_1)$	Unnormalized $\hat{P}(z_1, r_1)$	Normalized $\hat{P}(z_1\|r_1)$	Resampling
$p_1 =$ (0 , 0) $p_2 =$ (1 , 1) a_1, a_2	1/2 1/2	1/2 1	1/4 1/2	$4/3 * 1/4 = 1/3$ $4/3 * 1/2 = 2/3$	$p_1 =$ (1 , 1) $p_2 =$ (1 , 1) a_3, a_4

Transition $P(z_2 z_1)$	Belief $\hat{P}(z_2\|r_1)$	Weights $P(r_2 z_2)$	Unnormalized $\hat{P}(z_2, r_2\|r_1)$	Normalized $\hat{P}(z_2\|r_1, r_2)$	Resampling
$p_1 =$ (0 , 1) $p_2 =$ (1 , 1) a_5(left), a_6(stay)	1/2 1/2	1 1/2	1/2 1/4	$4/3 * 1/2 = 2/3$ $4/3 * 1/4 = 1/3$	$p_1 =$ (0 , 1) $p_2 =$ (1 , 1) a_7, a_8

Transition $P(z_3\|z_2)$	Belief $\hat{P}(z_3\|r_1, r_2)$	Weights $P(r_3\|z_3)$	Unnormalized $\hat{P}(z_3, r_3\|r_1, r_2)$	Normalized $\hat{P}(z_3\|r_1, r_2, r_3)$	Resampling
$p_1 =$ (-1 , 1) $p_2 =$ (1 , 1) a_9(left), a_{10}(stay)	1/2 1/2	1/4 1	1/8 1/2	$8/5 * 1/8 = 1/5$ $8/5 * 1/2 = 4/5$	$p_1 =$ (-1, 1) $p_2 =$ (1 , 1) a_{11}, a_{12}

(d) Use your samples (the unweighted particles in the last step) to evaluate the posterior probability that the x-coordinate of Z_3 is different than the column of Z_3, i.e. $X_3 \neq Y_3$.

Out of the two unweighted particles in the last step, exactly one satisfies $X_3 = Y_3$, so the estimate is 1/2.

(e) What is the problem of using the elimination algorithm instead of a particle filter for tracking Jabberwock?

The state space is infinite, so factors of infinite size (distributions over all points on the plane) would need to be computed and stored when using the variable elimination algorithm.