

# 1 Definitions: K-Means & Gaussian Mixture Models (GMM)

## 1.1 K-Means

For a visual walkthrough of K-means, see the [Clustering notes](#) on the course website.

1. **K-Means:** An unsupervised machine learning algorithm used for clustering data. The algorithm aims to partition a given dataset into K clusters, where K is a hyperparameter.

2. **K-Means Algorithm:**

Iteratively:

- (a) Assign data points to the closest cluster mean
- (b) Update the mean location based on the newly assigned points

This converges when the cluster assignments and the means no longer change.

3. **K-Means Optimization:** Our goal is to minimize the sum of the distances between each point and the center of its assigned cluster.

- Input:  $\{\mathbf{x}^{(i)}\}_{i=1}^N$  with  $\mathbf{x}^{(i)} \in \mathbb{R}^M$
- Output:  $z_i \in \{1, 2, \dots, K\}$ , the cluster assignment for each data point  $i$
- Output:  $\boldsymbol{\mu}_k \in \mathbb{R}^M$ , the center of cluster  $k$

The objective function is then:

$$\min_{\{z_i\}, \{\boldsymbol{\mu}_k\}} \sum_{i=1}^N \|\mathbf{x}^{(i)} - \boldsymbol{\mu}_{z_i}\|_2^2$$

4. **Alternating Minimization:** The objective above is actually quite difficult to optimize directly. Instead, we can use an alternating minimization approach.

- (a) **Fix cluster assignments, update cluster centers:** Given the cluster assignments, we can update the cluster centers by computing the mean of all data points assigned to each cluster.

$$\boldsymbol{\mu}_k = \frac{1}{N_k} \sum_{i=1}^N \mathbf{x}^{(i)} \mathbb{I}\{z_i = k\}$$

- (b) **Fix cluster centers, update cluster assignments:** Given the cluster centers, we can update the cluster assignments by assigning each data point to the cluster with the closest center.

$$z_i = \operatorname{argmin}_k \|\mathbf{x}^{(i)} - \boldsymbol{\mu}_k\|_2^2$$

## 1.2 Gaussian Mixture Models (GMM)

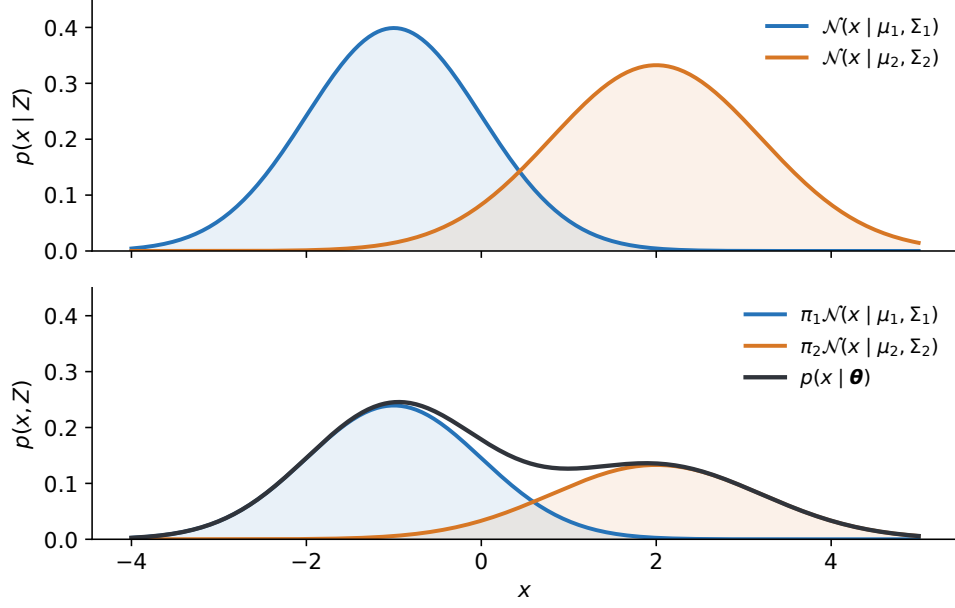
1. **GMM:** Probabilistic models used for clustering data. The algorithm assumes that the data is generated by a mixture of K Gaussian distributions, where K is a hyperparameter. GMM works by iteratively estimating the parameters of the Gaussian distributions and the weights of the mixture components using the Expectation-Maximization (EM) algorithm.
2. **EM:** An iterative method used for estimating the parameters of statistical models.

Let  $Z^{(i)}$  be a categorical random variable with components  $Z_1^{(i)}, Z_2^{(i)}, \dots, Z_K^{(i)}$ , where each component is 0 or 1, i.e.  $p(Z_k^{(i)} = 1 \mid \mathbf{x}^{(i)})$  is the probability that the  $i$ -th point  $\mathbf{x}^{(i)}$  belongs to the Gaussian distribution  $k$ .

Let  $\boldsymbol{\theta} = \boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \dots, \boldsymbol{\mu}_K, \Sigma_1, \dots, \Sigma_K, \pi_1, \dots, \pi_K$ , where  $\pi_k = p(Z_k^{(i)} = 1)$ .

The likelihood is  $\prod_{i=1}^N p(\mathbf{x}^{(i)} | \boldsymbol{\theta})$ .

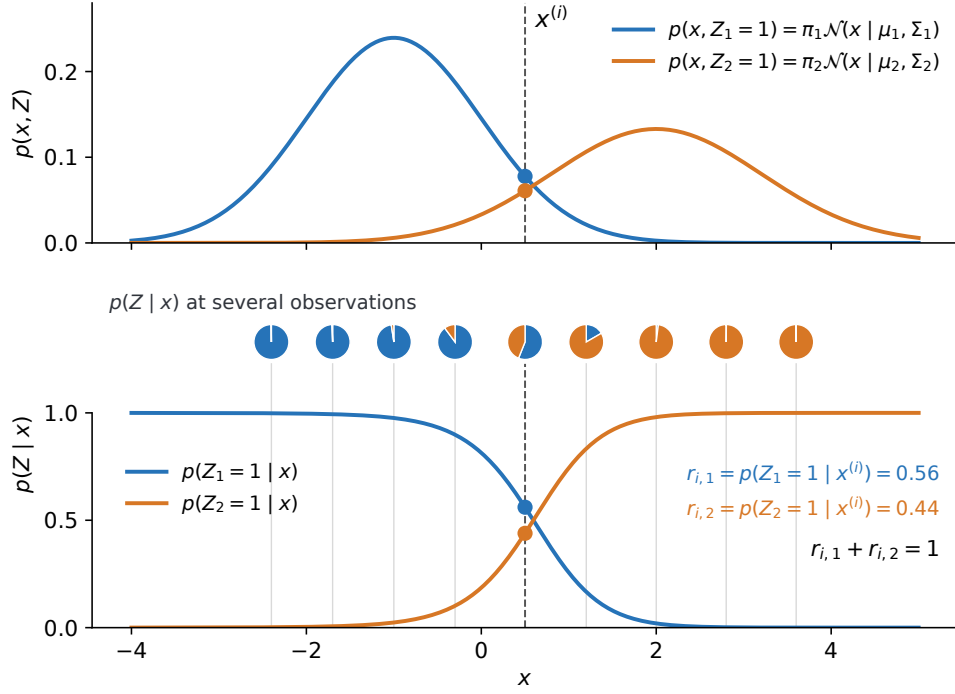
Hence, the log-likelihood is  $\ell(\boldsymbol{\theta}) = \sum_{i=1}^N \log p(\mathbf{x}^{(i)} | \boldsymbol{\theta}) = \sum_{i=1}^N \log \sum_{k=1}^K \pi_k \mathcal{N}(\mathbf{x}^{(i)} | \boldsymbol{\mu}_k, \Sigma_k)$ .



Top: the component densities  $p(x | Z_k = 1) = \mathcal{N}(x | \mu_k, \Sigma_k)$ . Bottom: weighting by  $\pi_k$  gives  $p(x, Z_k = 1)$ , and the weighted Gaussians sum to the density inside the logarithm,  $p(\mathbf{x}^{(i)} | \boldsymbol{\theta}) = \sum_{k=1}^K \pi_k \mathcal{N}(\mathbf{x}^{(i)} | \boldsymbol{\mu}_k, \Sigma_k)$ .

**E-Step:** Calculate the responsibilities  $r_{i,k} = p(Z_k^{(i)} = 1 | \mathbf{x}^{(i)}, \boldsymbol{\theta}) \forall i, k$ .

$$\begin{aligned}
 & p(Z_k^{(i)} = 1 | \mathbf{x}^{(i)}, \boldsymbol{\theta}) \\
 &= \frac{p(\mathbf{x}^{(i)} | Z_k^{(i)} = 1, \boldsymbol{\mu}_k, \Sigma_k) p(Z_k^{(i)} = 1 | \pi_k)}{p(\mathbf{x}^{(i)} | \boldsymbol{\theta})} \\
 &= \frac{\mathcal{N}(\mathbf{x}^{(i)} | \boldsymbol{\mu}_k, \Sigma_k) \pi_k}{\sum_{j=1}^K \pi_j \mathcal{N}(\mathbf{x}^{(i)} | \boldsymbol{\mu}_j, \Sigma_j)}
 \end{aligned}$$

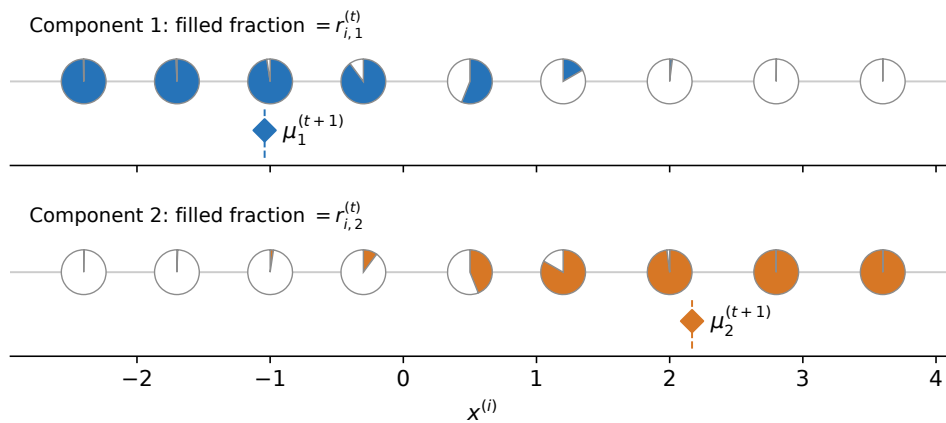


Top: the weighted densities  $p(x, Z_k = 1)$  at the observation  $x^{(i)}$  (dashed line). Bottom: dividing by their sum  $p(x | \theta)$  gives the posteriors  $p(Z_k = 1 | x)$  as functions of  $x$ ; the responsibilities  $r_{i,k}$  are their values at  $x^{(i)}$ , and  $\sum_k r_{i,k} = 1$  at every  $x$ . The pie charts show  $p(Z | x)$  at several observations: each one is split between the components in proportion to its responsibilities.

**M-Step:** Apply MLE and update the parameters  $\pi_k, \mu_k, \Sigma_k \forall k$ .  
Let's find the MLE for  $\mu_k$ .

$$\begin{aligned}
 \frac{\partial \ell}{\partial \mu_k} &= \frac{\partial}{\partial \mu_k} \sum_{i=1}^N \log \sum_{j=1}^K \pi_j \mathcal{N}(\mathbf{x}^{(i)} | \mu_j, \Sigma_j) \\
 &= \sum_{i=1}^N \frac{1}{\sum_{j=1}^K \pi_j \mathcal{N}(\mathbf{x}^{(i)} | \mu_j, \Sigma_j)} \frac{\partial}{\partial \mu_k} \sum_{j=1}^K \pi_j \mathcal{N}(\mathbf{x}^{(i)} | \mu_j, \Sigma_j) \\
 &= \sum_{i=1}^N \frac{1}{\sum_{j=1}^K \pi_j \mathcal{N}(\mathbf{x}^{(i)} | \mu_j, \Sigma_j)} \frac{\partial}{\partial \mu_k} \pi_k \mathcal{N}(\mathbf{x}^{(i)} | \mu_k, \Sigma_k) \\
 &= \sum_{i=1}^N \frac{\pi_k \mathcal{N}(\mathbf{x}^{(i)} | \mu_k, \Sigma_k)}{\sum_{j=1}^K \pi_j \mathcal{N}(\mathbf{x}^{(i)} | \mu_j, \Sigma_j)} \frac{\partial}{\partial \mu_k} \left( -\frac{1}{2} (\mathbf{x}^{(i)} - \mu_k)^\top \Sigma_k^{-1} (\mathbf{x}^{(i)} - \mu_k) \right) \\
 &= \sum_{i=1}^N p(Z_k^{(i)} = 1 | \mathbf{x}^{(i)}, \theta) \Sigma_k^{-1} (\mathbf{x}^{(i)} - \mu_k)
 \end{aligned}$$

We can set this to  $\mathbf{0}$ , and solve for  $\mu_k$  to get  $\mu_k = \frac{\sum_{i=1}^N p(Z_k^{(i)} = 1 | \mathbf{x}^{(i)}, \theta) \mathbf{x}^{(i)}}{\sum_{i=1}^N p(Z_k^{(i)} = 1 | \mathbf{x}^{(i)}, \theta)}$ .



Each observation contributes to each component's mean in proportion to its responsibility: the filled fraction of each pie is  $r_{i,k}^{(t)}$ ; diamonds mark the updated means.

We can do similar calculations for the other two parameters  $\pi_k$  and  $\Sigma_k$ .

$$\pi_k = \frac{\sum_{i=1}^N p(Z_k^{(i)}=1|\mathbf{x}^{(i)},\boldsymbol{\theta})}{N}$$

$$\Sigma_k = \frac{\sum_{i=1}^N p(Z_k^{(i)}=1|\mathbf{x}^{(i)},\boldsymbol{\theta})(\mathbf{x}^{(i)}-\boldsymbol{\mu}_k)(\mathbf{x}^{(i)}-\boldsymbol{\mu}_k)^\top}{\sum_{i=1}^N p(Z_k^{(i)}=1|\mathbf{x}^{(i)},\boldsymbol{\theta})}$$

## 2 K-Means

In KMeans, we aim to minimize the objective function:

$$J(K) = \min_{\mu_1, \dots, \mu_K, z_1, \dots, z_N} \sum_{i=1}^N \|\mathbf{x}^{(i)} - \mu_{z_i}\|_2^2$$

Recall the KMeans algorithm:

Let there be a set of  $N$  observations  $\mathcal{D} = \{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(N)}\}$ , where  $\mathbf{x}^{(i)} \in \mathbb{R}^M$  be the set of input examples that each have  $M$  features.

Initialize  $K$  cluster centers  $\{\mu_1, \dots, \mu_K\}$  where  $\mu_k \in \mathbb{R}^M$ .

Repeat until convergence:

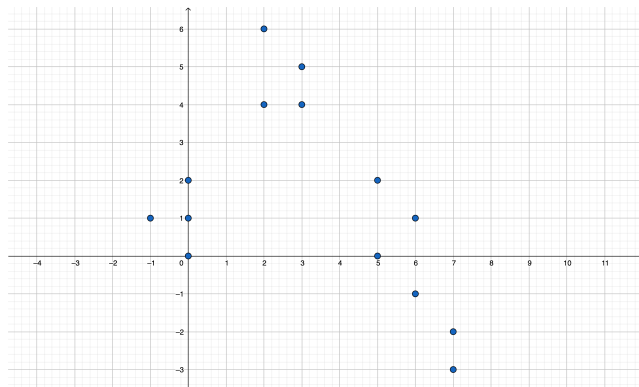
1. Assign each point  $\mathbf{x}^{(i)}$  to a cluster  $\mu_{z_i}$  where  $z_i = \operatorname{argmin}_{1 \leq k \leq K} \|\mathbf{x}^{(i)} - \mu_k\|_2^2$
2. Recompute each  $\mu_k$  as the mean of points assigned to it from the previous step.

What is the benefit of using k-means algorithm when solving a partitioning problem?

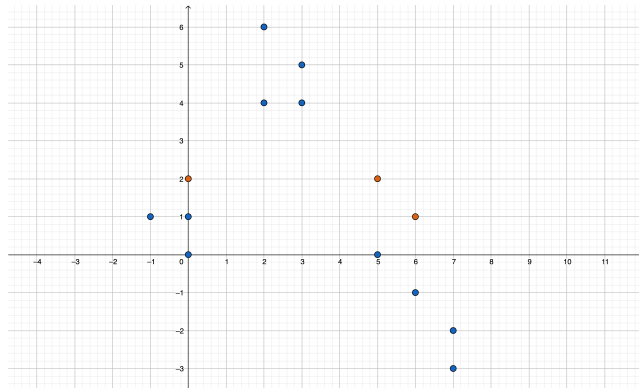
Brute-forcing (exhaustively enumerating all partitions) is NP-hard. Using K-means can solve the problem in  $O(lKN)$ , where  $l$  is the number of iterations,  $K$  is the number of cluster centers, and  $N$  is the number of points.

### 2.1 Walking through an example

Lets walk through an example of K-Means with  $k = 3$  using the following dataset for the first iteration:



Let the cluster centers be initialized to  $\mu_1 = (0, 2)$ ,  $\mu_2 = (5, 2)$ ,  $\mu_3 = (6, 1)$  as depicted below in the orange:



Perform one iteration of the KMeans algorithm:

1. What are the cluster assignments?

$$\begin{aligned}\mathcal{C}_1 &= \{(0, 0), (-1, 1), (0, 1), (0, 2), (2, 4), (2, 6)\} \\ \mathcal{C}_2 &= \{(3, 4), (3, 5), (5, 2)\} \\ \mathcal{C}_3 &= \{(5, 0), (6, 1), (6, -1), (7, -2), (7, -3)\}\end{aligned}$$

2. What are the recomputed cluster centers?

$$\begin{aligned}\mu_1 &= (0.5, 2.33) \\ \mu_2 &= (3.67, 3.67) \\ \mu_3 &= (6.2, -1)\end{aligned}$$

### 3 Gaussian Mixture Models vs. K-Means

What is the key difference between mixture modeling and K-means?

K-means is hard assignment (each point belongs to only one cluster) while mixture modeling is soft assignment (calculates probability that a point belongs to a cluster).

In GMM, we aim to maximize our likelihood. That is,  $\operatorname{argmax}_{\theta} \prod_{i=1}^N p(\mathbf{x}^{(i)} | \theta)$ .

$$\begin{aligned}\operatorname{argmax}_{\theta} \prod_{i=1}^N p(\mathbf{x}^{(i)} | \theta) &= \operatorname{argmax}_{\theta} \prod_{i=1}^N \sum_{k=1}^K p(\mathbf{x}^{(i)}, Z_k^{(i)} = 1 | \theta) \\ &= \operatorname{argmax}_{\theta} \prod_{i=1}^N \sum_{k=1}^K p(Z_k^{(i)} = 1) p(\mathbf{x}^{(i)} | Z_k^{(i)} = 1, \theta)\end{aligned}$$

What happens to this expression if we assume a hard-assignment? Simplify using the assumption.

Hard assignment means that  $p(Z_k^{(i)} = 1) = 1$  if the point belongs to the  $k$ th cluster.

Thus we have:

$$\operatorname{argmax}_{\boldsymbol{\theta}} \prod_{i=1}^N \sum_{k=1}^K p(Z_k^{(i)} = 1) p(\mathbf{x}^{(i)} \mid Z_k^{(i)} = 1, \boldsymbol{\theta})$$

Our points are from a gaussian distribution, thus we have:

$$\begin{aligned} & \operatorname{argmax}_{\boldsymbol{\theta}} \prod_{i=1}^N \sum_{k=1}^K p(Z_k^{(i)} = 1) \frac{1}{(2\pi\sigma^2)^{M/2}} \exp\left(\frac{-1}{2\sigma^2} \|\mathbf{x}^{(i)} - \boldsymbol{\mu}_k\|_2^2\right) \\ &= \operatorname{argmax}_{\boldsymbol{\theta}} \prod_{i=1}^N \exp\left(\frac{-1}{2\sigma^2} \|\mathbf{x}^{(i)} - \boldsymbol{\mu}_k\|_2^2\right) \end{aligned}$$

Taking the log, we get

$$\begin{aligned} &= \operatorname{argmax}_{\boldsymbol{\theta}} \log \prod_{i=1}^N \exp\left(\frac{-1}{2\sigma^2} \|\mathbf{x}^{(i)} - \boldsymbol{\mu}_k\|_2^2\right) \\ &= \operatorname{argmax}_{\boldsymbol{\theta}} \sum_{i=1}^N \log \exp\left(\frac{-1}{2\sigma^2} \|\mathbf{x}^{(i)} - \boldsymbol{\mu}_k\|_2^2\right) \\ &= \operatorname{argmax}_{\boldsymbol{\theta}} \sum_{i=1}^N \frac{-1}{2\sigma^2} \|\mathbf{x}^{(i)} - \boldsymbol{\mu}_k\|_2^2 \\ &= \operatorname{argmin}_{\boldsymbol{\mu}} \sum_{i=1}^N \|\mathbf{x}^{(i)} - \boldsymbol{\mu}_k\|_2^2 \end{aligned}$$

This looks familiar....it's K-means!

## 4 HMMs: Tracking a Jabberwock

You have been put in charge of a Jabberwock for your friend Lewis. The Jabberwock is kept in a large tulgey wood which is conveniently divided into an  $N \times N$  grid. It wanders freely around the  $N^2$  possible cells. At each time step  $t = 1, 2, 3, \dots$ , the Jabberwock is in some cell  $X_t \in \{1, \dots, N\}^2$ , 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  $\epsilon$ , 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  $P(X_2 = (4, 4))$ ?

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

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

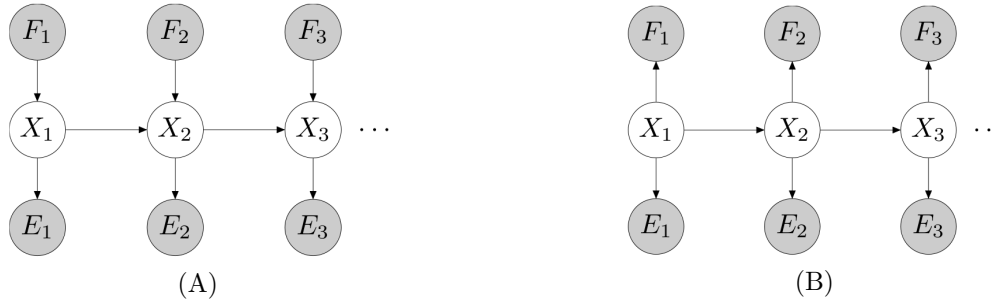
- (b) At each time step  $t$ , you don't see  $X_t$  but see  $E_t$ , which is the row that the Jabberwock is in; that is, if  $X_t = (r, c)$ , then  $E_t = r$ . Suppose we see that  $E_1 = 1$ ,  $E_2 = 2$ . Fill in the following table with the distribution over  $X_t$  after each time step, taking into consideration the evidence. Your answer should be concise. Hint: you should not need to do any heavy calculations.

| $t$ | $P(X_t   e_{1:t-1}, X_1 = (1, 1))$ |                                       | $P(X_t   e_{1:t}, X_1 = (1, 1))$ |                                           |
|-----|------------------------------------|---------------------------------------|----------------------------------|-------------------------------------------|
| 1   | $\mathbf{X_1}$                     | $\mathbf{P(X_1)}$                     | $\mathbf{X_1}$                   | $\mathbf{P(X_1)}$                         |
|     | (1, 1)                             |                                       | (1, 1)                           |                                           |
|     | all other values                   |                                       | all other values                 |                                           |
| 2   | $\mathbf{X_2}$                     | $\mathbf{P(X_2   e_1, X_1 = (1, 1))}$ | $\mathbf{X_2}$                   | $\mathbf{P(X_2   e_{1:2}, X_1 = (1, 1))}$ |
|     | (1, 2)                             |                                       | (2, 1)                           |                                           |
|     | (2, 1)                             |                                       | (2, $a$ ) ( $\forall a, a > 1$ ) |                                           |
|     | all other values                   |                                       | all other values                 |                                           |

| $t$ | $P(X_t   e_{1:t-1}, X_1 = (1, 1))$ |                                       | $P(X_t   e_{1:t}, X_1 = (1, 1))$ |                                           |
|-----|------------------------------------|---------------------------------------|----------------------------------|-------------------------------------------|
| 1   | $\mathbf{X_1}$                     | $\mathbf{P(X_1)}$                     | $\mathbf{X_1}$                   | $\mathbf{P(X_1)}$                         |
|     | (1, 1)                             | 1                                     | (1, 1)                           | 1                                         |
|     | all other values                   | 0                                     | all other values                 | 0                                         |
| 2   | $\mathbf{X_2}$                     | $\mathbf{P(X_2   e_1, X_1 = (1, 1))}$ | $\mathbf{X_2}$                   | $\mathbf{P(X_2   e_{1:2}, X_1 = (1, 1))}$ |
|     | (1, 2)                             | 51/200                                | (2, 1)                           | 51/60                                     |
|     | (2, 1)                             | 51/200                                | (2, $a$ ) ( $\forall a, a > 1$ ) | 1/60                                      |
|     | all other values                   | 1/200                                 | all other values                 | 0                                         |

At the first time step, we know the Jabberwock is at (1, 1) even without the evidence so it's probability is 1 and the other values are 0. At time 2, if we do not have evidence for that time step, all we know is that the Jabberwock was at position (1, 1) in row 1, at timestep 1. Using the same formulas as part a, we calculate the probabilities for each cell. If we do have evidence at the time step (right column), then we know the Jabberwock is on row 2. The probability the Jabberwock is on row 2 is  $1/4 + 1/20 = 3/10$ . Divide the probability of moving to (2,1) by the probability of ending in row 2 to get 51/60. For each of the other cells, since the probability must add to 1, they have a weight of 1/60.

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 add these frumious variables to the HMM. Consider the two candidates:



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

(A), because in the problem description the frumiousness  $F_t$  causes the teleport, so the choice of  $X_t$  depends on  $F_t$ .

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

- (d) 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 are given  $P(x_t, e_{1:t}, f_{1:t})$ . Multiplying by the transition CPT  $P(x_{t+1} | x_t, f_{t+1})$  and summing over  $x_t$  moves us forward one step. Then, to get the joint  $P(x_{t+1}, e_{1:t+1}, f_{1:t+1})$ , we multiply by the emission probability  $P(e_{t+1} | x_{t+1})$  and the CPT  $P(f_{t+1})$ .

- (e) 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.

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

Same idea as above, except the transition is now  $P(x_{t+1} | x_t)$ , since  $x_{t+1}$  no longer depends on  $f_{t+1}$ ; instead  $f_{t+1}$  is a second observation of  $x_{t+1}$ .

Now forget about the frumious variables and go back to the original HMM from (a) and (b). We will approximate the posterior from (b) with a particle filter that uses 4 particles  $p_1, \dots, p_4$ . Since  $X_1 = (1, 1)$  is known, all four particles start at  $(1, 1)$ , and the evidence  $E_1 = 1$  gives each of them weight 1, so nothing changes at  $t = 1$ .

As a source of randomness, use the values in order from the following sequence  $\{a_i\}_{1 \leq i \leq 10}$ :

|       |       |       |       |       |       |       |       |       |          |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|----------|
| $a_1$ | $a_2$ | $a_3$ | $a_4$ | $a_5$ | $a_6$ | $a_7$ | $a_8$ | $a_9$ | $a_{10}$ |
| 0.12  | 0.81  | 0.153 | 0.37  | 0.64  | 0.338 | 0.30  | 0.08  | 0.72  | 0.46     |

To sample from a discrete distribution  $P(X)$  over values ordered  $1, 2, \dots, n$  with  $a_i \sim U[0, 1)$ , return the value  $j$  such that  $\sum_{k=1}^{j-1} P(X = k) \leq a_i < \sum_{k=1}^j P(X = k)$ . Order grid cells row by row:  $(1, 1), (1, 2), \dots, (1, 10), (2, 1), \dots, (10, 10)$ .

**Elapse time.** For each particle in order  $p_1, p_2, p_3, p_4$ , take the next  $a_i$  and move the particle according to the table below. If it teleports, take the next  $a_i$  as well and sample the destination uniformly from the 100 cells, so that  $a_i \in [0.00, 0.01)$  gives  $(1, 1)$ ,  $[0.01, 0.02)$  gives  $(1, 2)$ , and so on.

|               |                  |
|---------------|------------------|
| $[0, 0.25)$   | Move to $(1, 2)$ |
| $[0.25, 0.5)$ | Move to $(2, 1)$ |
| $[0.5, 1)$    | Teleport         |

**Observe.** Weight each particle by  $P(E_2 = 2 \mid X_2)$ , which is 1 if the particle is in row 2 and 0 otherwise.

**Resample.** Draw 4 new particles from the normalized distribution, ordering its cells row by row as above.

- (f) Fill out the table using the particle filter described above. The  $a_i$ 's that you should use for each step are indicated in the last row of the table.

| Transition<br>$P(x_2 \mid x_1)$ | Belief<br>$\hat{P}(x_2 \mid e_1)$ | Weights<br>$P(e_2 \mid x_2)$ | Unnormalized<br>$\hat{P}(x_2, e_2 \mid e_1)$ | Normalized<br>$\hat{P}(x_2 \mid e_{1:2})$ | Resampling           |
|---------------------------------|-----------------------------------|------------------------------|----------------------------------------------|-------------------------------------------|----------------------|
| $p_1 = (1, 2)$                  | 1/4                               | 0                            | 0                                            | 0                                         | $p_1 = (2, 1)$       |
| $p_2 = (2, 6)$                  | 1/4                               | 1                            | 1/4                                          | 1/2                                       | $p_2 = (2, 1)$       |
| $p_3 = (2, 1)$                  | 1/4                               | 1                            | 1/4                                          | 1/2                                       | $p_3 = (2, 6)$       |
| $p_4 = (4, 4)$                  | 1/4                               | 0                            | 0                                            | 0                                         | $p_4 = (2, 1)$       |
| $a_1, \dots, a_6$               |                                   |                              |                                              |                                           | $a_7, \dots, a_{10}$ |

**Elapse:**  $a_1 = 0.12 < 0.25$ , so  $p_1$  moves to  $(1, 2)$ .  $a_2 = 0.81 \geq 0.5$ , so  $p_2$  teleports, and  $a_3 = 0.153 \in [0.15, 0.16)$  is cell number 16 in row-by-row order, which is  $(2, 6)$ .  $a_4 = 0.37 \in [0.25, 0.5)$ , so  $p_3$  moves to  $(2, 1)$ .  $a_5 = 0.64$ , so  $p_4$  teleports, and  $a_6 = 0.338 \in [0.33, 0.34)$  is cell 34, which is  $(4, 4)$ .

**Weight:** the four particles are in different cells, so each has belief 1/4. Only  $p_2$  and  $p_3$  are in row 2, so they keep their belief and the others drop to 0. Normalizing gives 1/2 on  $(2, 1)$  and 1/2 on  $(2, 6)$ .

**Resample:** in row-by-row order  $(2, 1)$  comes before  $(2, 6)$ , so  $[0, 0.5) \rightarrow (2, 1)$  and  $[0.5, 1) \rightarrow (2, 6)$ . The values  $a_7 = 0.30$ ,  $a_8 = 0.08$ ,  $a_9 = 0.72$ ,  $a_{10} = 0.46$  give  $(2, 1), (2, 1), (2, 6), (2, 1)$ .

- (g) What is the particle filter's estimate of  $P(X_2 = (2, 1) \mid e_{1:2})$  after resampling, and how does it compare with your exact answer from (b)?

Three of the four resampled particles are at  $(2, 1)$ , so  $\hat{P}(X_2 = (2, 1) \mid e_{1:2}) = 3/4$ , compared with the exact  $51/60 \approx 0.85$ . With only 4 particles the estimate is coarse (it can only be a multiple of  $1/4$ ); it approaches the exact answer as the number of particles grows.

- (h) Suppose instead that all four particles had landed outside row 2 after the elapse step. What goes wrong, and what is one way to make this less likely?

Every particle gets weight 0, so the weights cannot be normalized and there is nothing to resample from: the filter has lost track of the Jabberwock (particle depletion). It happens here because the sensor is exact, so any particle in the wrong row gets zero weight. Using more particles makes it unlikely that none of them lands in row 2; a noisier sensor model that gives every row some nonzero probability also avoids zero weights, and in the worst case we can reinitialize the particles uniformly over the cells consistent with the evidence.