Sharif university of technology
Faculty of electrical engineering

Neuroscience ,learning ,memory and cognition

Dr. karbalaie

Soraya Charkas
99101387

HomeWork 4

June 6, 2023

Part one : practical

Brain regions in dataset

We write the code as below to categorize the brain regions:

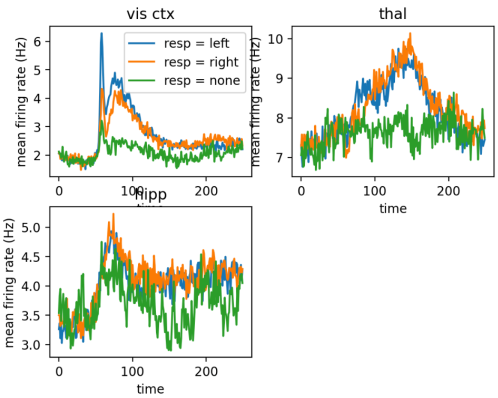
```
In [15]: nareas = 3
         Neurons = len(dat['brain_area'])
         barea = nareas * np.ones(Neurons, )
         for j in range(nareas):
           barea[np.isin(dat['brain_area'], brain_groups[j])] = j
         print(barea)
```

```
[0. 0. 0. 2. 0. 0. 2. 2. 0. 0. 2. 2. 0. 0. 2. 0. 0. 0. 2. 2. 2. 0. 2. 0.
 0. 0. 2. 0. 2. 2. 2. 2. 2. 2. 2. 0. 0. 2. 0. 2. 2. 0. 2. 2. 2. 1. 2. 0.
 0. 2. 2. 0. 2. 2. 2. 0. 2. 2. 0. 0. 2. 0. 1. 0. 2. 2. 2. 2. 2. 2. 0. 1.
 2. 2. 2. 2. 2. 2. 2. 1. 2. 2. 2. 2. 2. 2. 2. 2. 0. 0. 2. 2. 2. 0. 2. 0.
 2. 2. 2. 2. 0. 2. 2. 2. 2. 2. 0. 2. 2. 0. 0. 2. 2. 2. 2. 2. 2. 2. 0. 2.
 0. 0. 0. 0. 2. 2. 2. 2. 0. 0. 1. 2. 2. 0. 2. 0. 0. 0. 1. 2. 2. 0. 2. 2.
 2. 2. 2. 2. 2. 2. 0. 2. 2. 2. 1. 2. 2. 2. 1. 2. 2. 2. 2. 2. 2. 2. 2. 2.
 2. 2. 2. 1. 2. 0. 2. 2. 2. 2. 1. 2. 0. 0. 1. 0. 0. 2. 2. 2. 2. 2. 0. 2.
 2. 0. 2. 0. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 0. 2. 2. 2. 0. 2. 2. 0. 0.
 2. 2. 2. 2. 0. 2. 0. 0. 3. 3. 3. 3. 3. 3. 3. 3. 3. 3. 3. 3. 3. 3. 3. 3.
 3. 3. 3. 3. 3. 3. 3. 3. 3. 3. 3. 3. 3. 3. 3. 3. 3. 3. 3. 3. 3. 3. 3. 3.
 3. 3. 3. 3. 3. 3. 3. 3. 3. 3. 3. 3. 3. 3. 3. 3. 3. 3. 3. 3. 3. 3. 3. 3.
 3. 3. 3. 3. 3. 3. 3. 3. 3. 3. 3. 3. 3. 3. 3. 3. 3. 3. 3. 3. 3. 3. 3. 3.
 3. 3. 3. 3. 3. 3. 3. 3. 3. 3. 3. 3. 3. 3. 3. 3. 3. 3. 3. 3. 3. 3. 3. 3.
 3. 3. 3. 3. 3. 3. 3. 3. 3. 3. 3. 3. 3. 3. 3. 3. 3. 3. 3. 3. 3. 3. 3. 3.
 3. 3. 3. 3. 3. 3. 3. 3. 3. 3. 3. 3. 3. 3. 3. 3. 3. 3. 3. 3. 3. 3. 3. 3.
 3. 3. 3. 3. 3. 3. 3. 3. 3. 3. 3. 3. 3. 3. 3. 3. 3. 3. 2. 2. 0. 1. 2. 0.
 1. 1. 1. 0. 0. 0. 2. 2. 1. 1. 1. 1. 1. 0. 0. 0. 0. 0. 0. 1. 1. 1. 1. 1.
 0. 0. 0. 2. 1. 2. 2. 2. 2. 0. 0. 2. 2. 1. 1. 0. 0. 1. 1. 0. 1. 2. 2. 2.
 2. 2. 0. 1. 2. 1. 0. 0. 2. 1. 0. 1. 1. 2. 1. 0. 1. 1. 1. 1. 2. 0. 1. 1.
 2. 1. 2. 1. 0. 0. 2. 2. 0. 1. 1. 1. 2. 2. 2. 2. 0. 1. 0. 1. 1. 0. 0. 1.
 1. 0. 1. 2. 2. 0. 1. 0. 2. 2. 1. 0. 1. 2. 0. 1. 2. 1. 2. 0. 2. 0. 0. 1.
 1. 0. 1. 1. 2. 1. 1. 0. 0. 0. 2. 1. 1. 1. 2. 0. 1. 1. 1. 2. 0. 0. 1. 1.
 1. 0. 2. 1. 1. 0. 0. 1. 2. 1. 0. 1. 2. 1. 0. 2. 1. 2. 0. 0. 1. 1. 1. 1.
 1. 1. 1. 1. 1. 0. 2. 1. 1. 0. 2. 1. 1. 0. 0. 1. 1. 2. 1. 2. 2. 1. 2. 1.
 1. 1. 1. 1. 1. 1. 1. 1. 0. 1. 1. 1. 1. 1. 2. 1. 1. 1. 2. 1. 0. 2. 1. 1.
 1. 1. 1. 1. 0. 0. 1. 0. 1. 0. 2. 1. 0. 0. 1. 1. 1. 0. 1. 1. 0. 1. 2. 1.
 1. 1. 2. 1. 2. 2. 0. 1. 1. 0. 2. 0. 1. 2. 1. 2. 1. 1. 0. 1. 1. 0. 2. 1.
 1. 1. 2. 0. 2. 2. 1. 1. 1. 2. 1. 0. 2. 0. 0. 2. 2. 1. 2. 1. 1. 1. 0. 0.
 2. 1.]
```

Plot by brain regions

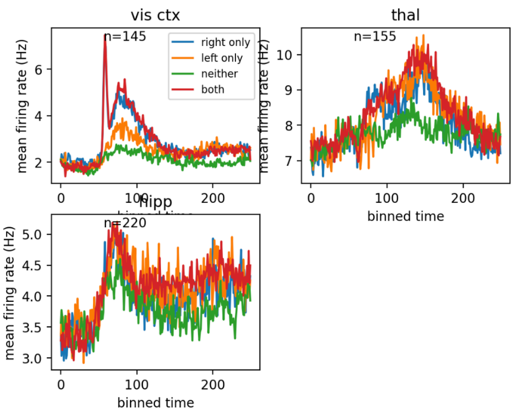
As we done in previous homework, we plot the brain region activity for vis ctx , thal and hip with respect to rat response type:

```
dt = dat['bin_size']  # binning at 10 ms
NT = dat['spks'].shape[-1]

regions = ["vis ctx", "thal", "hipp", "other ctx", "midbrain", "basal ganglia", "cortical subplate", "other"]
region_colors = ['blue', 'red', 'green', 'darkblue', 'violet', 'lightblue', 'orange', 'gray']
response = dat['response']  # right - nogo - left (-1, 0, 1)
for j in range(nareas):
  ax = plt.subplot(2, 2, j + 1)
  plt.title(regions[j])
  if np.sum(barea == j) == 0:
    continue
  plt.plot(1/dt * dat['spks'][barea == j][:, response < 0].mean(axis=(0, 1)))
  plt.plot(1/dt * dat['spks'][barea == j][:, response > 0].mean(axis=(0, 1)))
  plt.plot(1/dt * dat['spks'][barea == j][:, response == 0].mean(axis=(0, 1)))

  if j == 0:
    plt.legend(['resp = left', 'resp = right', 'resp = none'], fontsize=10)
  ax.set(xlabel='time', ylabel='mean firing rate (Hz)')
plt.show()
```



And here we plot with respect to visual conditions:

```
vis_left = dat['contrast_left']  # 0 - low - high
vis_right = dat['contrast_right']  # 0 - low - high
for j in range(nareas):
  ax = plt.subplot(2, 2, j + 1)

  plt.plot(1/dt * dat['spks'][barea==j][:, np.logical_and(vis_left == 0, vis_right > 0)].mean(axis=(0, 1)))
  plt.plot(1/dt * dat['spks'][barea==j][:, np.logical_and(vis_left > 0, vis_right == 0)].mean(axis=(0, 1)))
  plt.plot(1/dt * dat['spks'][barea==j][:, np.logical_and(vis_left == 0, vis_right == 0)].mean(axis=(0, 1)))
  plt.plot(1/dt * dat['spks'][barea==j][:, np.logical_and(vis_left > 0, vis_right > 0)].mean(axis=(0, 1)))
  plt.text(.25, .92, 'n=%d'%np.sum(barea == j), transform=ax.transAxes)

  if j==0:
    plt.legend(['right only', 'left only', 'neither', 'both'], fontsize=8)
  ax.set(xlabel='binned time', ylabel='mean firing rate (Hz)', title=regions[j])
plt.show()
```



What can you infer by comparing these six figures? alo compare regions, it seems that hippocampus is somehow moving average of visual cortex? find some correlation between regions and discuss in your report

Answer:

By comparing the six figures depicting brain region activity in relation to rat response type and visual conditions, we can make several inferences. Firstly, when comparing the visual cortex (vis ctx) and hippocampus (hipp) activity, it appears that there is a correlation between these regions. Specifically, the hippocampus seems to exhibit a similar pattern to the moving average of the visual cortex. This correlation suggests that there may be an interaction or information exchange between the visual cortex and the hippocampus. It is possible that the hippocampus receives input from the visual cortex and integrates or processes this information in some way.

Now we compare what is the difference withvrespect to rat response type and visual condition:

1. Visual Condition:

The visual cortex shows significant activity variations based on different visual conditions. This suggests that it plays a crucial role in processing visual information. The hippocampus and thalamus may also exhibit some degree of activity modulation based on visual conditions, although to a lesser extent compared to the visual cortex.

2. Rat Response Type:

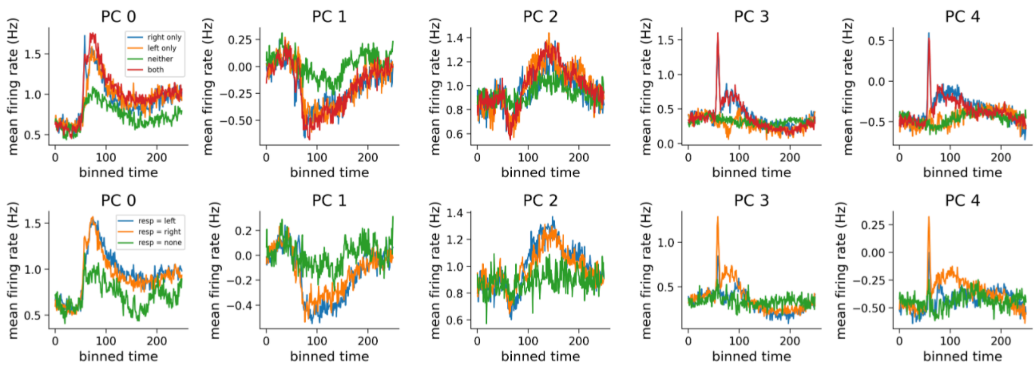
The visual cortex displays distinct patterns of activity depending on the rat's response type. This indicates that it is involved in processing and differentiating responses to visual stimuli. The hippocampus and thalamus might also exhibit some response-related activity, albeit less prominently than the visual cortex.

Overall, these observations imply that all three regions contribute to the processing of visual information and the generation of responses in rats. The visual cortex appears to play a central role in visual processing, while the hippocampus and thalamus may have complementary functions in this process.

PCA (Principle Component Analysis)

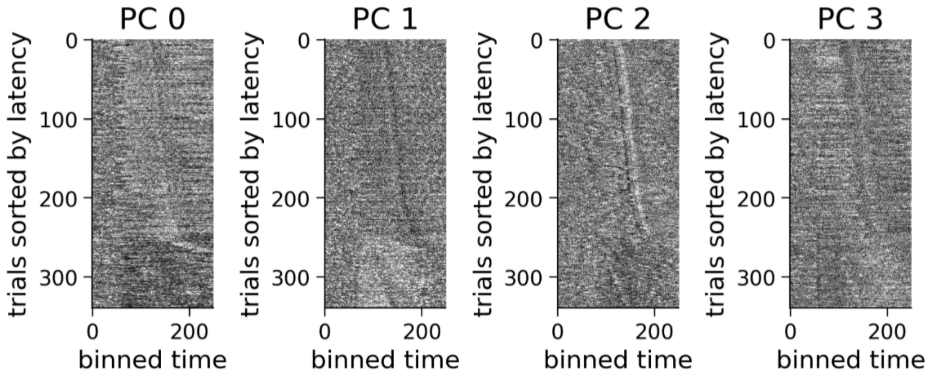
We write the code as said, the first line selects 51-130 bins from the data then we reshape the selected data, and after that we run the PCA function. The rest of the code is just plotting the datas which we selected using Matplotlib's syntaxes

```
data = np.reshape(dat['spks'][:, :, 51:130], (NN, -1))  # first 80 bins = 1.6 sec
data = data - np.mean(data, axis=1)[:, np.newaxis]
model = PCA(n_components=5).fit(droll.T)
W = model.components_
pc = W @ np.reshape(dat['spks'], (NN, -1))
pc = np.reshape(pc, (5, -1, NT))

plt.figure(figsize= (20, 6))
for j in range(len(pc)):
  ax = plt.subplot(2, len(pc) + 1, j + 1)
  pc1 = pc[j]

  plt.plot(pc1[np.logical_and(vis_left == 0, vis_right > 0), :].mean(axis=0))
  plt.plot(pc1[np.logical_and(vis_left > 0, vis_right == 0), :].mean(axis=0))
  plt.plot(pc1[np.logical_and(vis_left == 0, vis_right == 0), :].mean(axis=0))
  plt.plot(pc1[np.logical_and(vis_left > 0, vis_right > 0), :].mean(axis=0))

  if j == 0:
    plt.legend(['right only', 'left only', 'neither', 'both'], fontsize=8)
  ax.set(xlabel = 'binned time', ylabel='mean firing rate (Hz)')
  plt.title('PC %d'%j)

  ax = plt.subplot(2, len(pc) + 1, len(pc) + 1 + j + 1)

  plt.plot(pc1[response > 0, :].mean(axis=0))
  plt.plot(pc1[response < 0, :].mean(axis=0))
  plt.plot(pc1[response == 0, :].mean(axis=0))

  if j == 0:
    plt.legend(['resp = left', 'resp = right', 'resp = none'], fontsize=8)
  ax.set(xlabel='binned time', ylabel='mean firing rate (Hz)')
  plt.title('PC %d'%j)
plt.show()
```



We also do the next part by using the argsort function and then plot the desired figures:

```
isort = np.argsort(dat['response_time'].flatten())
for j in range(4):
  ax = plt.subplot(2,4, j+1 )
  pc1 = zscore(pc[j])
  plt.imshow(pc1[isort, :], aspect='auto', vmax=2, vmin=-2, cmap='gray')
  ax.set(xlabel='binned time', ylabel='trials sorted by latency')
  plt.title('PC %d'%j)
plt.show()
```



Each PC represents a different linear combination of neural activity patterns across the trials. To infer the concept behind each PC, we can examine the weights or loadings of the neurons on that PC. The weights indicate the contribution of each neuron to the PC.

Part two : theory

Deciding where the fish go

1. Answer:

When you have an equal chance, it's better to fish on the left side because there are no submarines there, so you'd pick that spot.

2. Answer:

If there's a good chance of finding fish on the right side, then go for it! The high probability of fish being on the right side is way more important than the slightly better fishing on the left side (which you probably won't even get).

3. Answer:

If there's just a slightly better chance of finding fish on the right side compared to the left, still go for the left side because you'll have a better overall experience there. Remember, it's not just about picking the side with higher chances - the utility holds significant importance.

Likelihood of the fish being on either side

Think! 3: Guessing the location of the fish

1. Answer :

The fisher-person is on the left side so:

- P(m = fish | s = left) = 0.7 because they have a 70% chance of catching a fish when on the same side as the school

- P(m = no fish | s = left) = 0.3 because the probability of catching a fish and not catching a fish for a given state must add up to 1 as these are the only options: 1 - 0.7 = 0.3

- P(m = fish | s = right) = 0.2

- P(m = no fish | s = right) = 0.8

2.Answer:

If the fisher catches a fish, it's more likely that the school of fish is on the left side. That's because the chances of catching a fish when the school is on the left side which is 0.7 are higher than when it's on the right side which is 0.2.

3.Answer:

If the fisher-person doesn't catch anything, you'd probably think the fish are on the right side. That's because the chances of not catching anything when the fish are on the right side which is 0.8 are higher than when they're on the left side which is 0.3 .

Correlation and marginalization

Think! 4.1: Covarying probability distributions

1.Answer:

When the correlation is zero, the two components have nothing in common and are completely independent. this means you can not get any information about a variable by observing the other one .Importantly, the marginal distribution of one variable is therefore independent of the other.

2.Answer:

The correlation determines how the probabilities are spread out in the joint probability table. When the correlation is stronger, the probabilities are more constrained because both rows and columns must add up to one! The marginal probabilities indicate the relative importance, but as the correlation approaches 1 or -1, the absolute probabilities for one characteristic rely more on the other characteristic.

3.Answer:

The correlation determines how much chance there is for stuff to be on the diagonals. When the correlation gets close to 1 (or -1), the probability of seeing either of the two pairings has to get really low.

4.Answer:

If we think about what information we gain by observing one quality, the intuition from (3.) tells us that we know more (have more information) about the other quality as a function of the correlation.

Math Exercise 4.2.1: Computing marginal probabilities

1.Answer:

The probability of a fish being silver is the joint probability of it being small and silver plus the joint probability of it being large and silver:

P(Y = silver) = P(X = small, Y = silver) + P(X = large, Y = silver) = 0.4 + 0.1 =0.5

2.Answer:

The probability of such state is 1. Because the fish should be either silver or gold(there are no other ways)

3.Answer:

The marginal probabilities:

P(X = small) = P(X = small, Y = silver) + P(X = small, Y = gold) = 0.6

P(Y = gold) = P(X = small, Y = gold) + P(X = large, Y = gold) = 0.5

We already know the joint probability: P(X = small, Y = gold) = 0.2

We can now use the given formula:

P(X = small or Y = gold) = P(X = small) + P(Y = gold) - P(X = small, Y =gold)

= 0.6 + 0.5 - 0.2 = 0.9

Math Exercise 4.2.2: Computing marginal likelihood

1.Answer:

P(m = fish) = P(m = fish, s = left) + P(m = fish, s = right) =

P(m = fish | s left)P(s = left) + P(m = fish | s = right)P(s = right)

= 0.1 * 0.3 + .5 * .7 = 0.38

2.Answer:

P(m = fish) = P(m = fish, s = left) + P(m = fish, s = right) =

P(m = fish | s = left)P(s = left) + P(m = fish | s = right)P(s = right)

= 0.1 * 0.6 + .5 * .4 = 0.26

Baye's Rule and the Posterior

Math Exercise 5: Calculating a posterior probability

1.Answer:

Using Baye's rule, we know that P(s = left | m = fish) = P(m = fish | s = left)P(s = left) / P(m = fish)

Let's first compute P(m = fish):

P(m = fish) = P(m = fish | s = left)P(s = left) + P(m = fish | s = right) P(s = right) = 0.5 * 0.3 + .1 * .7 = 0.22

Now we can plug in all parts of Baye's rule:

P(s = left | m = fish) = P(m = fish | s = left)P(s = left) / P(m = fish) = 0.5 * 0.3 / 0.22 = 0.68

2.Answer:

Using Baye's rule, we know that P(s = right | m = no fish) = P(m = no fish | s = right)P(s = right) / P(m = no fish)

Let's first compute P(m = no fish):

P(m = no fish) = P(m = no fish | s = left)P(s = left) + P(m = no fish | s = right)P(s = right) = 0.5 * 0.3 + 0.9 *0 .7 = 0.78

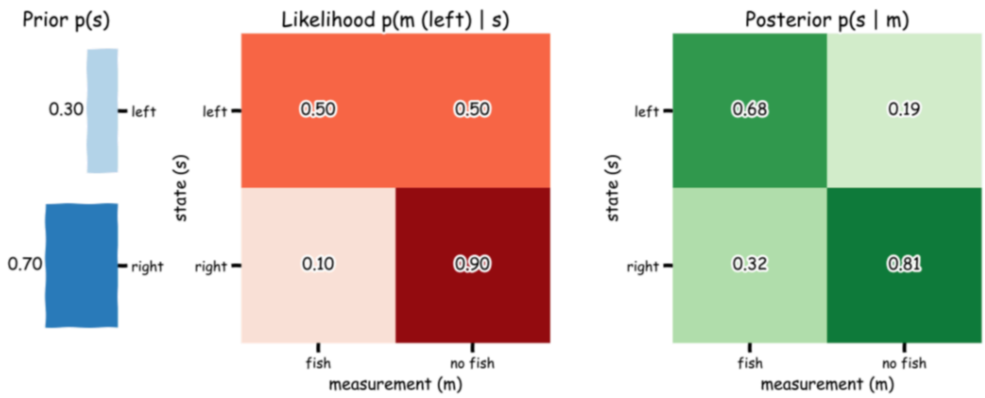
Now we can plug in all parts of Baye's rule:

P(s = right | m = no fish) = P(m = no fish | s = right)P(s = right) / P(m = no fish) = 0.9 * 0.7 / 0.78 = 0.81

Extra (Bonus): Coding Exercise 5: Computing Posteriors

To implement this section, we only have to implement the Baye's rule using python.the completed code and the output is as below :

In [88]:
```
def compute_posterior(likelihood, prior):
  """ Use Bayes' Rule to compute posterior from likelihood and prior

  Args:
    likelihood (ndarray): i x j array with likelihood probabilities where i is
                number of state options, j is number of measurement options
    prior (ndarray): i x 1 array with prior probability of each state

  Returns:
    ndarray: i x j array with posterior probabilities where i is
           number of state options, j is number of measurement options

  """

  # Compute unnormalized posterior (likelihood times prior)
  posterior = likelihood * prior # first row is s = left, second row is s = right

  # Compute p(m)
  p_m = np.sum(posterior, axis = 0)

  # Normalize posterior (divide elements by p_m)
  posterior /= p_m

  return posterior


# Make prior
prior = np.array([0.3, 0.7]).reshape((2, 1)) # first row is s = left, second row is s =

# Make likelihood
likelihood = np.array([[0.5, 0.5], [0.1, 0.9]]) # first row is s = left, second row is

# Compute posterior
posterior = compute_posterior(likelihood, prior)

# Visualize
with plt.xkcd():
  plot_prior_likelihood_posterior(prior, likelihood, posterior)
```



Extra (Bonus): Interactive Demo 5: What affects the posterior?

1.Answer:

The initial guess has a big impact on the final guess if it gives us a lot of useful info: like when we strongly believe that the fish are mostly on one side or the other. If we have a really high initial belief that the fish are on the left side (let's say 90% sure), then no matter what, our final guess will lean heavily towards the left side.

2.Answer:

When the chances are about the same, catching or not catching a fish doesn't give you much useful information. Basically, if you're equally likely to catch a fish no matter where you are, it doesn't tell you anything important! The differences in the probabilities show how much useful info you can actually get.

3.Answer:

Just like before, the chances have the biggest say when they provide useful information. When catching a fish gives you a lot of clues about which side is more likely to have fish. For instance, if the fisher catches nothing when they fish on the right side and the fish school is on the left (p fish | s = left) = 0, but catches a fish when the school is on the right side (p fish | s = right) = 1, then the initial probabilities don't matter anymore. The catch reveals the whole hidden truth.

Section 6: Making Bayesian fishing decisions

Interactive Demo! 6: What is more important, the probabilities or the utilities?

1.Answer:

There are actually tons of different combinations that can give you the same expected outcome for both actions. However, the probabilities you calculate afterwards have to balance out the variations in the usefulness. So, what really matters is that, with a particular usefulness measure, there's always a point where you don't really care which option you choose.

2.Answer:

What really counts is the relative information: if the chances are pretty even, then the likelihood matters more. If the measurement is equally likely to go either way (meaning it doesn't tell you much), then what you believed before becomes more significant. But the key idea from Bayes Rule and the Bayesian approach is that what truly matters is the relative information you get from a measurement, and you can take all of this information into account when making your decision.

3.Answer:

The model shows us a clear method to think about how we're supposed to put information together and what actions we should take, based on our goals. In this situation, assuming our goal is to get the most out of our choices, we can say what an animal or person ought to do.

4.Answer:

There are lots of possible extensions. Humans may not always try to maximize utility; humans and animals might not be able to calculate or represent probability distributions exactly; The utility function might be more complicated.

The end