HOMEWORK 6

>>Thong Nguyen<<
>>9084850198<<

Instructions: Use this latex file as a template to develop your homework. We are changing our reproducibility policy on code submissions going forward. **Instead of uploading it on GitHub, please submit a separate zip file that contains your code. You will submit two files to Canvas, one is your pdf, and the other one is a zip file.** Late submissions may not be accepted. You can choose any programming language (i.e. python, R, or MATLAB). Please check Piazza for updates about the homework.

1 Implementation: GAN (30 pts)

In this part, you are expected to implement GAN with MNIST dataset. We have provided a base jupyter notebook (gan-base.ipynb) for you to start with, which provides a model setup and training configurations to train GAN with MNIST dataset.

(a) Implement training loop and report learning curves and generated images in epoch 1, 50, 100. Note that drawing learning curves and visualization of images are already implemented in provided jupyter notebook. (15 pts)

Procedure 1 Training GAN, modified from **?**

```
Input: $m$: real data batch size, $n_z$: fake data batch size
Output: Discriminator $D$, Generator $G$
  for number of training iterations do
      # Training discriminator
      Sample minibatch of $n_z$ noise samples $\{z^{(1)}, z^{(2)}, \cdots, z^{(n_z)}\}$ from noise prior $p_g(z)$
      Sample minibatch of $\{x^{(1)}, x^{(2)}, \cdots, x^{(m)}\}$
      Update the discriminator by ascending its stochastic gradient:
          $$\nabla_{\theta_d}\big(\frac{1}{m}\sum_{i=1}^{m}\log D(x^{(i)}) + \frac{1}{n_z}\sum_{i=1}^{n_z}\log(1 - D(G(z^{(i)})))\big)$$
      # Training generator
      Sample minibatch of $n_z$ noise samples $\{z^{(1)}, z^{(2)}, \cdots, z^{(n_z)}\}$ from noise prior $p_g(z)$
      Update the generator by ascending its stochastic gradient:
          $$\nabla_{\theta_g}\frac{1}{n_z}\sum_{i=1}^{n_z}\log D(G(z^{(i)}))\big)$$
  end for
  # The gradient-based updates can use any standard gradient-based learning rule. In the base code, we are using
  Adam optimizer (?)
```

Expected results are as follows.

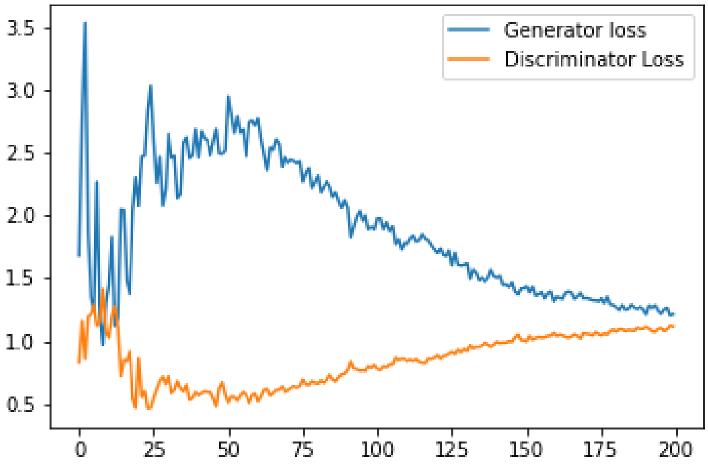
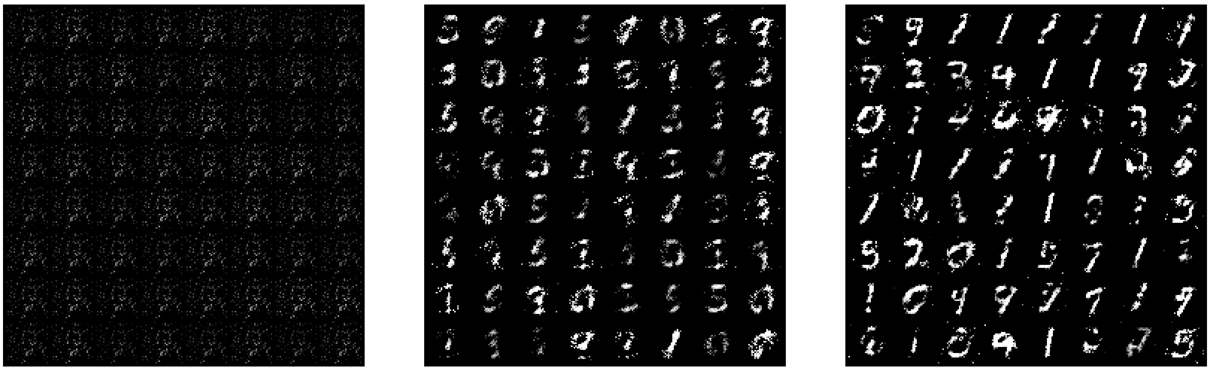


Figure 1: Learning curve



(a) epoch 1

(b) epoch 50

(c) epoch 100

Figure 2: Generated images by G

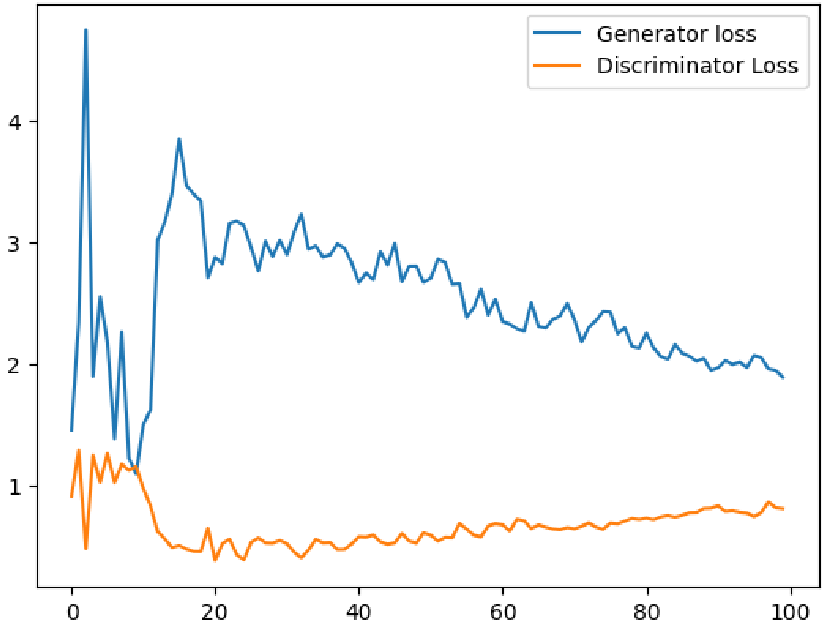
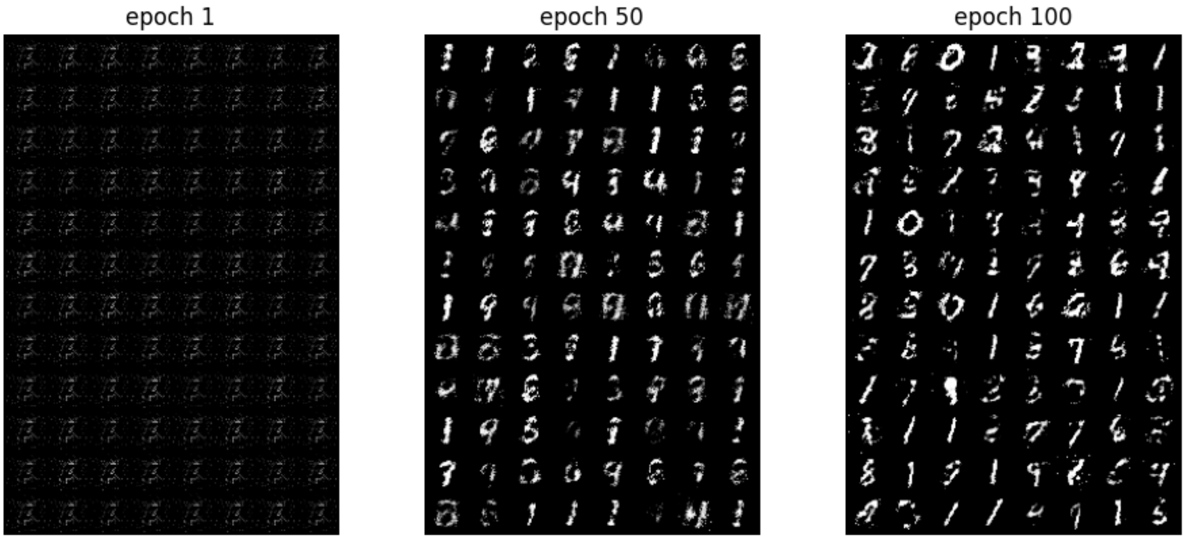


Figure 3: Learning curve



(a) epoch 1

(b) epoch 50

(c) epoch 100

Figure 4: Generated images by G

(b) Replace the generator update rule as the original one in the slide,
"Update the generator by descending its stochastic gradient:

$$\nabla_{\theta_g} \frac{1}{n_z} \sum_{i=1}^{n_z} \log(1 - D(G(z^{(i)})))$$

" , and report learning curves and generated images in epoch 1, 50, 100. Compare the result with (a). Note that it may not work. If training does not work, explain why it doesn't work. (10 pts)

Explain why this doesn't work: The generator is updated based stochastic gradient that encourages the discriminator to classify the fake since its loss is calculated using the negative log-likelihood of the discriminator's output for the generated fake sample being "fake." In other words, this equation for stochastic gradient tries to minimize the probability of the discriminator classifying the generated data as real. Therefore, the generator failed to fool the discriminator, and the discriminator could easily reject the sample.

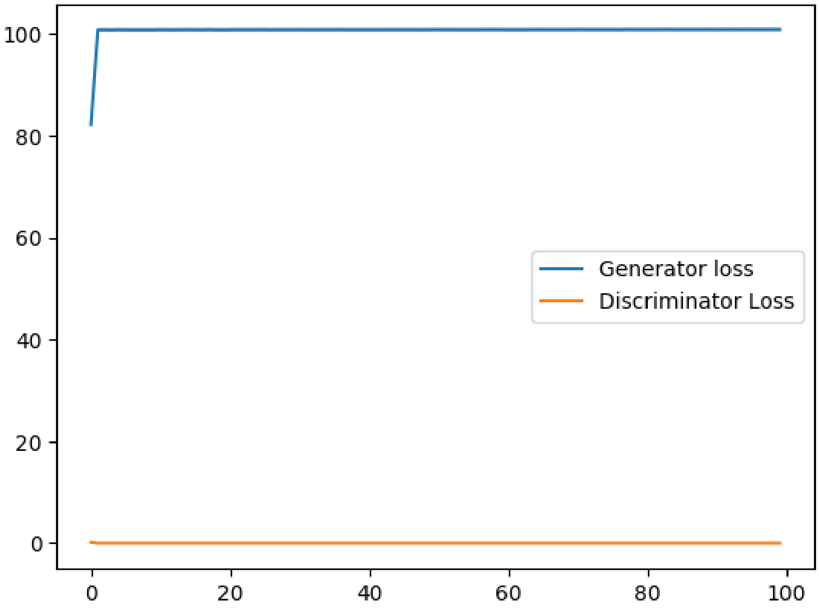
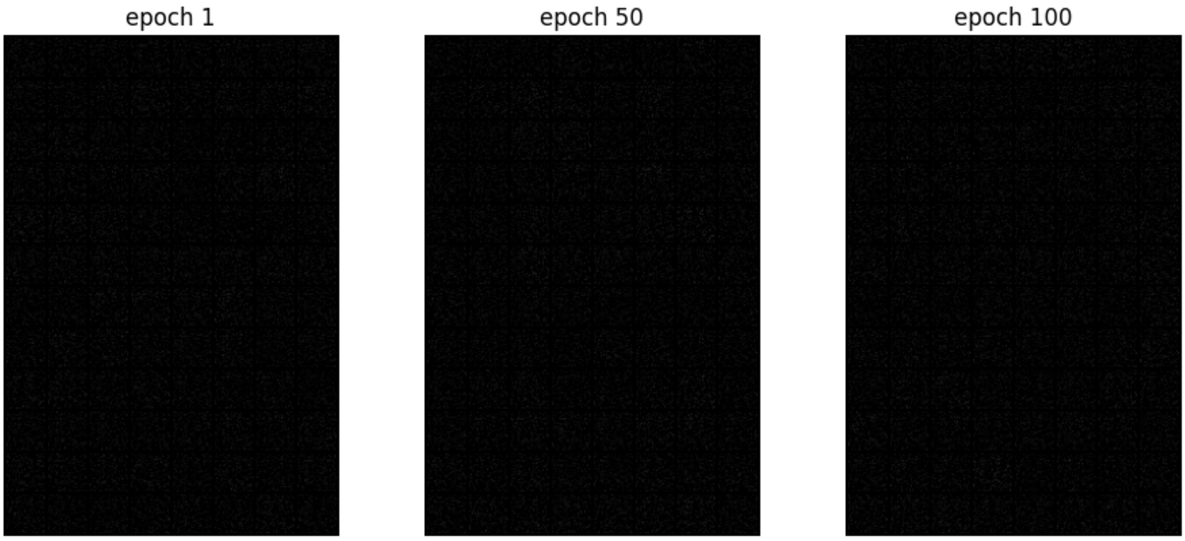


Figure 5: Learning curve



(a) epoch 1

(b) epoch 50

(c) epoch 100

Figure 6: Generated images by G

(c) Except the method that we used in (a), how can we improve training for GAN? Implement that and report your setup, learning curves, and generated images in epoch 1, 50, 100. (5 pts)

Setup: Using one-sided label smoothing (0.2), we apply one-sided label smoothing only to the real labels in the discriminator loss. Earlier, label/target values for a classifier were 0 or 1; 0 for fake images and 1 for real images. This means we can have decimal values such as 0.9 (true), 0.8 (true), 0.1 (fake), or 0.2 (fake) instead of labeling every example as either 1 (true) or 0 (fake). We smooth the target values (label values) of the real and fake images. Label smoothing can reduce the risk of adversarial examples in GANs. This can make the discriminator more robust to adversarial examples. Among that, adding some noise by applying an occasion switch of labels for the discriminator and swapping real and fake labels can help the discriminator to be stable and robust, preventing it from getting too strong. Also, learning rate schedule reducing the learning rate during training gradually can help the model to converge to a better solution.

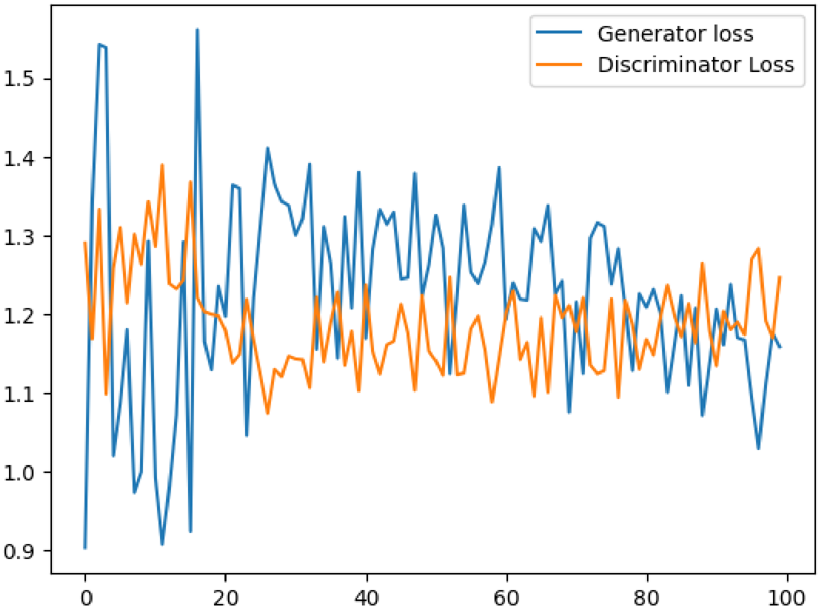
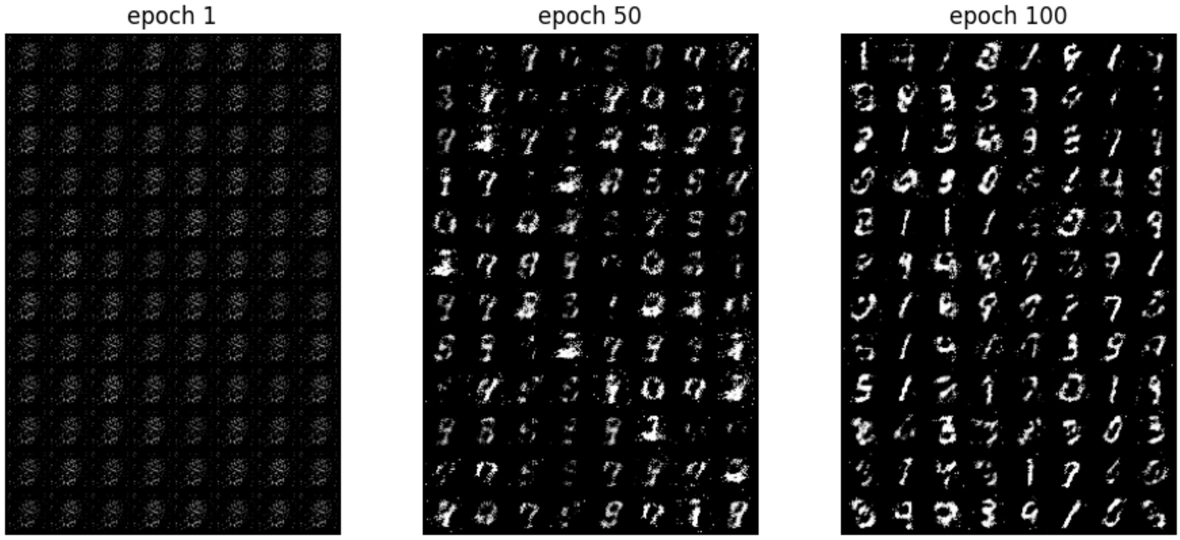


Figure 7: Learning curve



(a) epoch 1

(b) epoch 50

(c) epoch 100

2 Review change of variables in probability density functions [25 pts]

In Flow based generative model, we have seen $p_\theta(x) = p(f_\theta(x))|\frac{\partial f_\theta(x)}{\partial x}|$. As a hands-on (fixed parameter) example, consider the following setting.

Let X and Y be independent, standard normal random variables. Consider the transformation $U = X + Y$ and $V = X - Y$. In the notation used above, $U = g_1(X, Y)$ where $g_1(X, Y)$ where $g_1(x, y) = x + y$ and $V = g_2(X, Y)$ where $g_2(x, y) = x - y$. The joint pdf of X and Y is $f_{X,Y} = (2\pi)^{-1} exp(-x^2/2) exp(-y^2/2), -\infty < x < \infty, -\infty < y < \infty$. Then, we can determine u, v values by x, y, i.e. $\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$.

(a) Compute Jacobian matrix

$$J = \begin{bmatrix} \frac{\partial x}{\partial u} & \frac{\partial x}{\partial v} \\ \frac{\partial y}{\partial u} & \frac{\partial y}{\partial v} \end{bmatrix}$$

(5 pts)

We have $U = g_1(x, y) = x + y, V = g_2(x, y) = x - y \to x = (U + V)/2, y = (U - V)/2 \to \frac{\partial x}{\partial u} = \frac{1}{2}, \frac{\partial x}{\partial v} = \frac{1}{2}, \frac{\partial y}{\partial u} = \frac{1}{2}, \frac{\partial y}{\partial v} = -\frac{1}{2},$

J = $\begin{bmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & -\frac{1}{2} \end{bmatrix}$

(b) (Forward) Show that the joint pdf of U, V is

$$f_{U,V}(u, v) = \big(\frac{1}{\sqrt{2\pi}\sqrt{2}} exp(-u^2/4)\big)\big(\frac{1}{\sqrt{2\pi}\sqrt{2}} exp(-v^2/4)\big)$$

(10 pts)

(Hint: $f_{U,V}(u, v) = f_{X,Y}(?, ?)|det(J)|$)

$$|det(J)| = | -\frac{1}{4} - \frac{1}{4}| = \frac{1}{2}$$

So, we can write the joint PDF of U and V as:

$$\begin{aligned} f_{U,V}(u, v) &= f_{X,Y}\left(\frac{1}{2}(u + v), \frac{1}{2}(u - v)\right) \cdot \frac{1}{2} \\ &= \frac{1}{2\pi} \exp\left(-\frac{(u + v)^2}{8}\right) \exp\left(-\frac{(u - v)^2}{8}\right) \cdot \frac{1}{2} \\ &= \frac{1}{4\pi} \exp\left(-\frac{u^2}{4}\right) \exp\left(-\frac{v^2}{4}\right). \\ &= \big(\frac{1}{\sqrt{2\pi}\sqrt{2}} \exp(-u^2/4)\big)\big(\frac{1}{\sqrt{2\pi}\sqrt{2}} \exp(-v^2/4)\big) \end{aligned}$$

(c) (Inverse) Check whether the following equation holds or not.

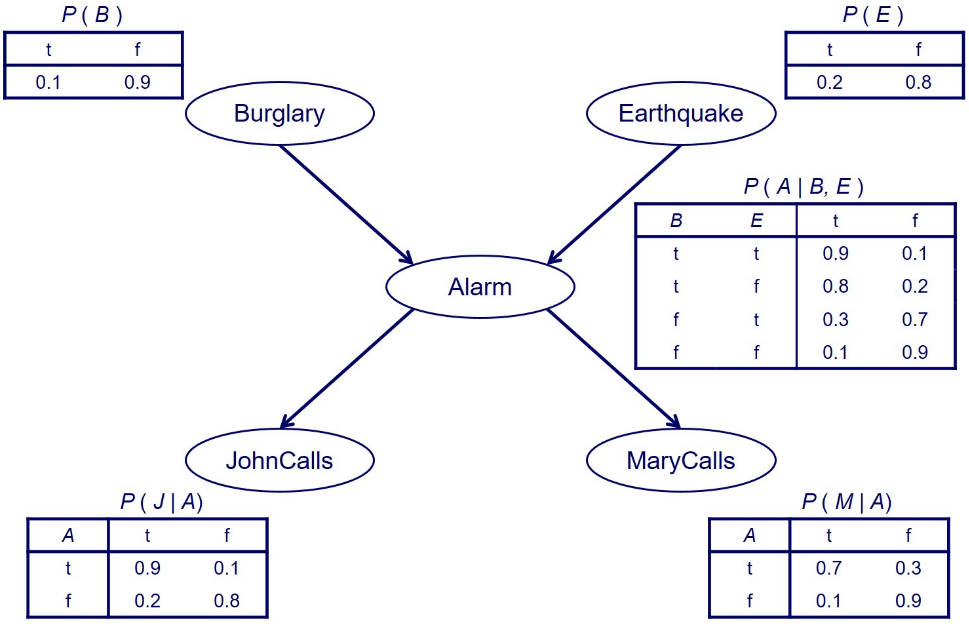
$$f_{X,Y}(x, y) = f_{U,V}(x + y, x - y)|det(J)^{-1}|$$

(10 pts)

$$\begin{aligned} f_{U,V}(x + y, x - y) &= \big(\frac{1}{\sqrt{2\pi}\sqrt{2}} \exp(-(x + y)^2/4)\big)\big(\frac{1}{\sqrt{2\pi}\sqrt{2}} \exp(-(x - y)^2/4)\big)(\frac{1}{2})^{-1} \\ &= \frac{1}{2\pi} \exp(-(x + y)^2/4) \exp(-(x - y)^2/4) \\ &= \frac{1}{2\pi} \exp(-x^2/2) \exp(-y^2/2) \\ &= f_{X,Y}(x, y) \end{aligned}$$

3 Directed Graphical Model [20 points]

Consider the directed graphical model (aka Bayesian network) in Figure 9.



P (B)

t	f
0.1	0.9

P (E)

t	f
0.2	0.8

P (A | B, E)

B	E	t	f
t	t	0.9	0.1
t	f	0.8	0.2
f	t	0.3	0.7
f	f	0.1	0.9

P (J | A)

A	t	f
t	0.9	0.1
f	0.2	0.8

P (M | A)

A	t	f
t	0.7	0.3
f	0.1	0.9

Figure 9: A Bayesian Network example.

Compute $P(B = t \mid E = f, J = t, M = t)$ and $P(B = t \mid E = t, J = t, M = t)$. (10 points for each) These are the conditional probabilities of a burglar in your house (yikes!) when both of your neighbors John and Mary call you and say they hear an alarm in your house, but without or with an earthquake also going on in that area (what a busy day), respectively.

$P(B = t, E = f, J = t, M = t) = P(J = t, M = t|A)P(A|B = t, E = f)P(B = t)P(E = f) = P(J = t|A)P(M = t|A)P(A|B = t, E = f)P(B = t)P(E = f) = (0.1)(0.8)(0.8)(0.9)(0.7) + (0.1)(0.8)(0.2)(0.2)(0.1) = 0.04064$

$P(E = f, J = t, M = t) = P(B = t, E = f, J = t, M = t) + P(B = f, E = f, J = t, M = t)$
$= P(B = t, E = f, A = t, J = t, M = t) + P(B = t, E = f, A = f, J = t, M = t) + P(B = f, E = f, J = t, M = t, A = t) + P(B = f, E = f, J = t, M = t, A = f)$
$= (0.1)(0.8)(0.8)(0.9)(0.7)+(0.1)(0.8)(0.2)(0.2)(0.1)+(0.9)(0.8)(0.1)(0.9)(0.7)+(0.9)(0.8)(0.9)(0.2)(0.1) = 0.09896$

$$P(B = t|E = f, J = t, M = t) = \frac{P(B = t, E = f, J = t, M = t)}{P(E = f, J = t, M = t)} = \frac{0.04064}{0.09896} = 0.410670978$$

=============

$P(B = t, E = t, J = t, M = t) = P(J = t, M = t|A)P(A|B = t, E = t)P(B = t)P(E = t) = P(J = t|A)P(M = t|A)P(A|B = t, E = t)P(B = t)P(E = t) = (0.1)(0.2)(0.9)(0.9)(0.7)+(0.1)(0.2)(0.1)(0.2)(0.1) = 0.01138$

$P(E = t, J = t, M = t) = P(B = t, E = t, J = t, M = t) + P(B = f, E = t, J = t, M = t)$
$= P(B = t, E = t, A = t, J = t, M = t) + P(B = t, E = t, A = f, J = t, M = t) + P(B = f, E = t, J = t, M = t, A = t) + P(B = f, E = t, J = t, M = t, A = f)$
$= (0.1)(0.2)(0.9)(0.9)(0.7)+(0.1)(0.2)(0.1)(0.2)(0.1)+(0.9)(0.2)(0.3)(0.9)(0.7)+(0.9)(0.2)(0.7)(0.2)(0.1) = 0.04792$

$$P(B = t|E = t, J = t, M = t) = \frac{P(B = t, E = t, J = t, M = t)}{P(E = t, J = t, M = t)} = \frac{0.01138}{0.04792} = 0.237479132$$

4 Chow-Liu Algorithm [25 pts]

Suppose we wish to construct a directed graphical model for 3 features X, Y, and Z using the Chow-Liu algorithm. We are given data from 100 independent experiments where each feature is binary and takes value T or F. Below is a table summarizing the observations of the experiment:

X	Y	Z	Count
T	T	T	36
T	T	F	4
T	F	T	2
T	F	F	8
F	T	T	9
F	T	F	1
F	F	T	8
F	F	F	32

Compute the joint probabilities $P(X, Y)$:

$$P(X = T, Y = T) = \frac{36 + 4}{100} = 0.4$$
$$P(X = T, Y = F) = \frac{8 + 2}{100} = 0.1$$
$$P(X = F, Y = T) = \frac{9 + 1}{100} = 0.1$$
$$P(X = F, Y = F) = \frac{32 + 8}{100} = 0.4$$

Compute the joint probabilities $P(X, Z)$:

$$P(X = T, Z = T) = \frac{36 + 2}{100} = 0.38$$
$$P(X = T, Z = F) = \frac{4 + 8}{100} = 0.12$$
$$P(X = F, Z = T) = \frac{9 + 8}{100} = 0.17$$
$$P(X = F, Z = F) = \frac{32 + 1}{100} = 0.33$$

Compute the joint probabilities $P(Y, Z)$:

$$P(Y = T, Z = T) = \frac{36 + 9}{100} = 0.45$$
$$P(Y = T, Z = F) = \frac{4 + 1}{100} = 0.05$$
$$P(Y = F, Z = T) = \frac{2 + 8}{100} = 0.1$$
$$P(Y = F, Z = F) = \frac{32 + 8}{100} = 0.4$$

Compute the marginal probabilities:

$$P(X = T) = \frac{36 + 4 + 2 + 8}{100} = 0.5$$
$$P(X = F) = \frac{9 + 1 + 8 + 32}{100} = 0.5$$
$$P(Y = T) = \frac{36 + 4 + 9 + 1}{100} = 0.5$$
$$P(Y = F) = \frac{2 + 8 + 8 + 32}{100} = 0.5$$
$$P(Z = T) = \frac{36 + 2 + 9 + 8}{100} = 0.55$$
$$P(Z = F) = \frac{4 + 8 + 1 + 32}{100} = 0.45$$

1. Compute the mutual information $I(X, Y)$ based on the frequencies observed in the data. (5 pts)
   $I(X,Y) = \sum\sum P(x,y) \times \log \frac{P(x,y)}{P(x) \times P(y)} = 0.4 \times \log(\frac{0.4}{0.5 \times 0.5}) + 0.1 \times \log(\frac{0.1}{0.5 \times 0.5}) + 0.1 \times \log(\frac{0.1}{0.5 \times 0.5}) + 0.4 \times \log(\frac{0.4}{0.5 \times 0.5}) = 0.278$

2. Compute the mutual information $I(X, Z)$ based on the frequencies observed in the data. (5 pts)
   $I(X,Z) = \sum\sum P(x,z) \times \log \frac{P(x,z)}{P(x) \times P(z)} = 0.38 \times \log(\frac{0.38}{0.5 \times 0.55}) + 0.12 \times \log(\frac{0.12}{0.5 \times 0.45}) + 0.17 \times \log(\frac{0.17}{0.5 \times 0.55}) + 0.33 \times \log(\frac{0.33}{0.5 \times 0.45}) = 0.1136$

3. Compute the mutual information $I(Z, Y)$ based on the frequencies observed in the data. (5 pts)
   $I(Y,Z) = \sum\sum P(y,z) \times \log \frac{P(y,z)}{P(x) \times P(z)} = 0.45 \times \log(\frac{0.45}{0.5 \times 0.55}) + 0.05 \times \log(\frac{0.05}{0.5 \times 0.45}) + 0.1 \times \log(\frac{0.1}{0.5 \times 0.55}) + 0.4 \times \log(\frac{0.4}{0.5 \times 0.45}) = 0.3232$

4. Which undirected edges will be selected by the Chow-Liu algorithm as the maximum spanning tree? (5 pts)
   To find the maximum spanning tree, we will choose the edges with the highest mutual information values, which are $I(X,Y) = 0.278$ and $I(Y,Z) = 0.3232$

5. Root your tree at node X, assign directions to the selected edges. (5 pts)
   To root the tree at node X, we assign directions to the edges from the root to other nodes:
   $X \rightarrow Y$
   $Y \rightarrow Z$
   So, the directed tree will be: $X \rightarrow Y \rightarrow Z$.