HOMEWORK 6

APOORVA KUMAR
908 461 5997

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 Goodfellow et al. (2014)

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 (Kingma and Ba, 2014)

Expected results are as follows.

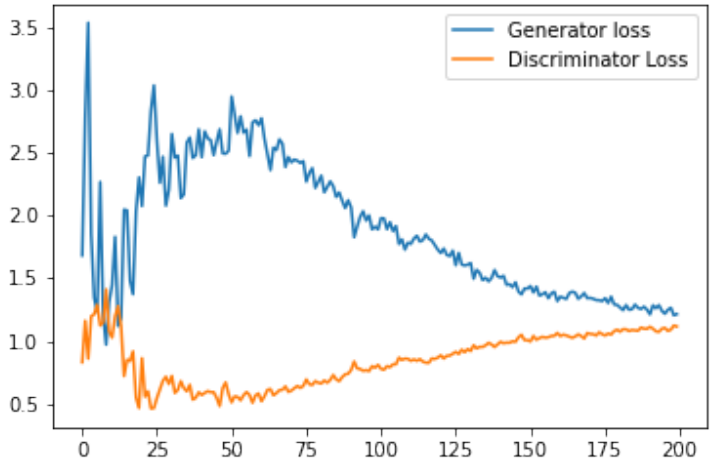
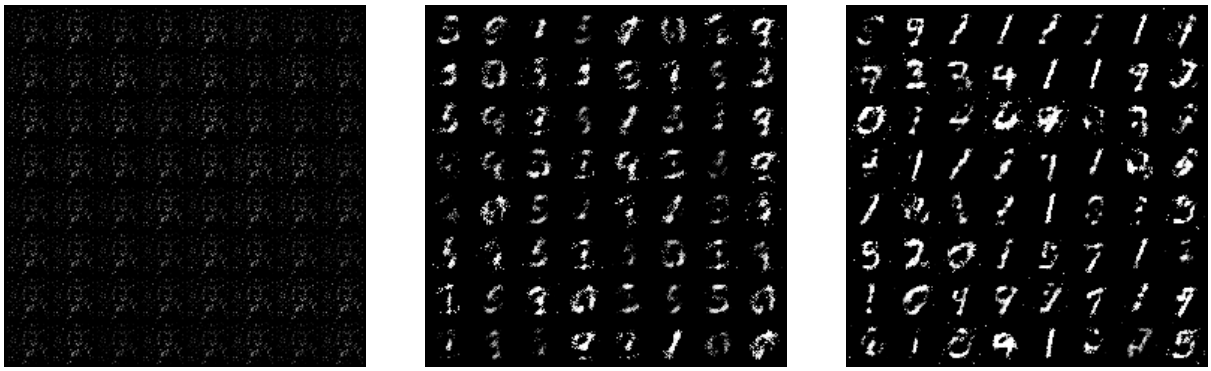


Figure 1: Learning curve



(a) epoch 1 (b) epoch 50 (c) epoch 100

Figure 2: Generated images by G

Solution begins here:

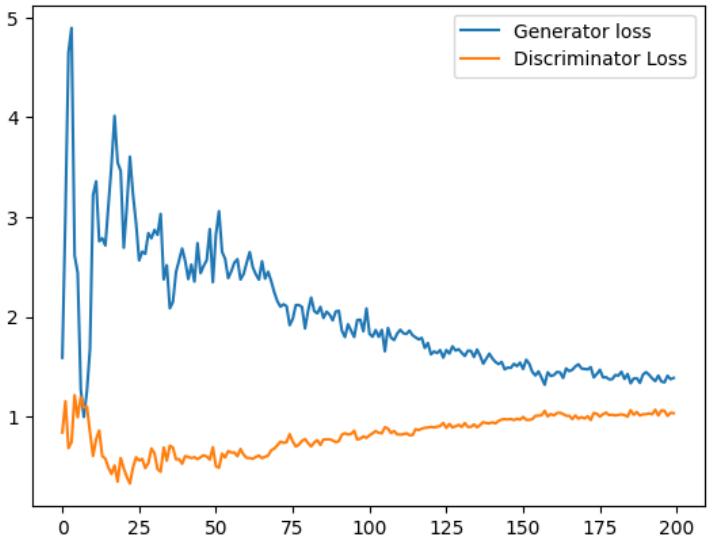
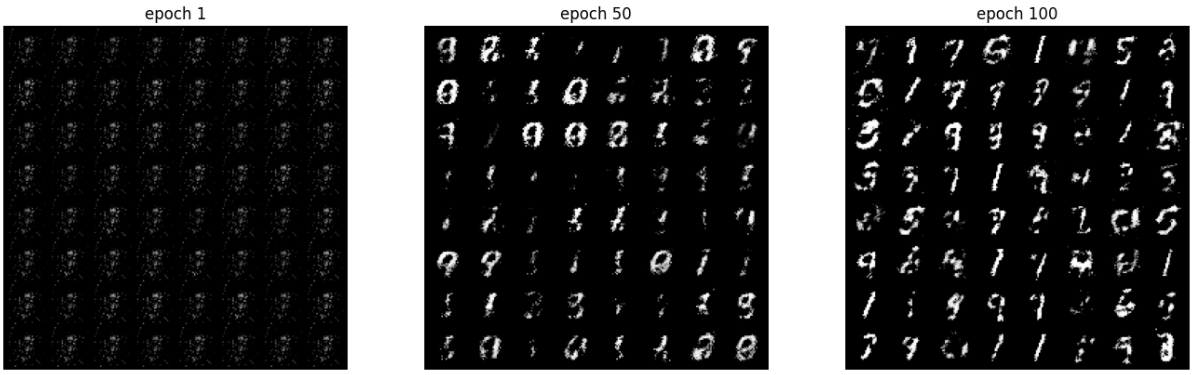


Figure 3: Solution Learning curve



(a) Epoch 1 (b) Epoch 50 (c) Epoch 100

Figure 4: Solution 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:

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

" , 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)

Here we see that the training doesn't work because the gradients start to saturate early during the training causing the Discriminator to reject every sample out of the Generator because of how the loss function works. In part (a) we maximize $\log D(G(z))$ which provides us with enough opposite directional gradient during early training for the Generator to move in the right direction. On the other hand during the training of this part we see that loss function can be brought down easily just by the Discriminator thus providing no gradient to the Generator to learn anything. The same can also be seen on the learning curve which hits zero instantaneously.

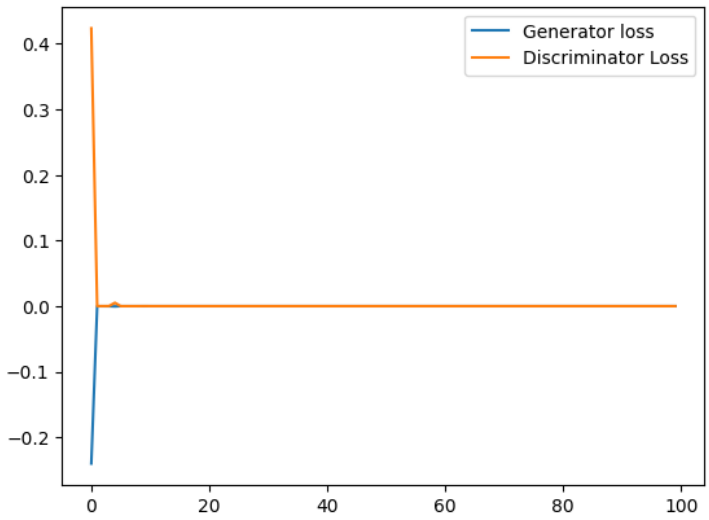
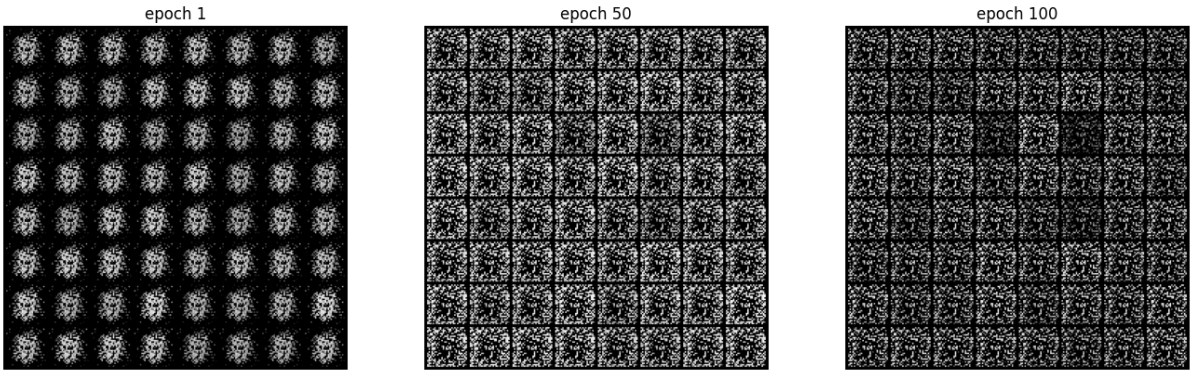


Figure 5: Solution Learning curve



(a) Epoch 1 (b) Epoch 50 (c) Epoch 100

Figure 6: Solution 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)

We know that the main task of Generator is to fool the Discriminator into thinking its a true sample. Due to the way neural networks work its very easy to learn certain feature relation to classify and make decisions while neglecting other features in the set. The Generator can easily exploit this issue and build a model which achieves this relation without caring about the values of the other features. Thus our major task is to make the Discriminator more robust which inturn will make the Generator better. In an attempt to achieve this I set the true labels to 0.5 rather than 1 with a probability of 0.4 confusing the Discriminator. Its like an implicit dropout to force the Discriminator to learn better and make better connections. While this might result in a worse learning curve the output of the Generator can be seen to be much better than part (a).

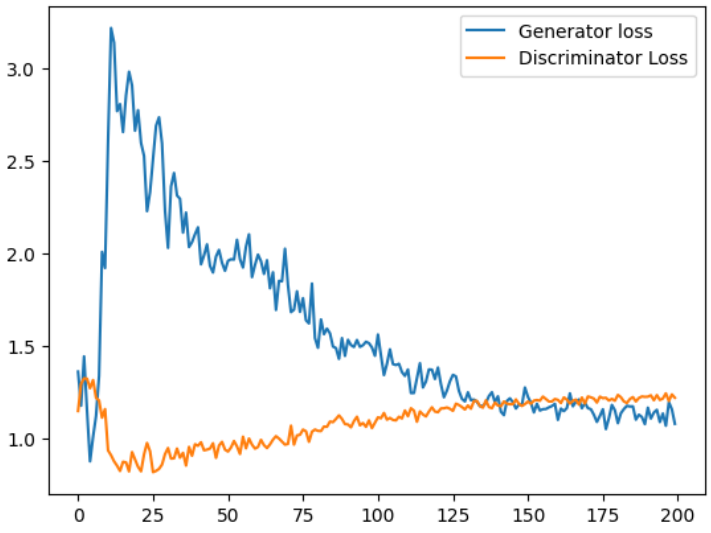
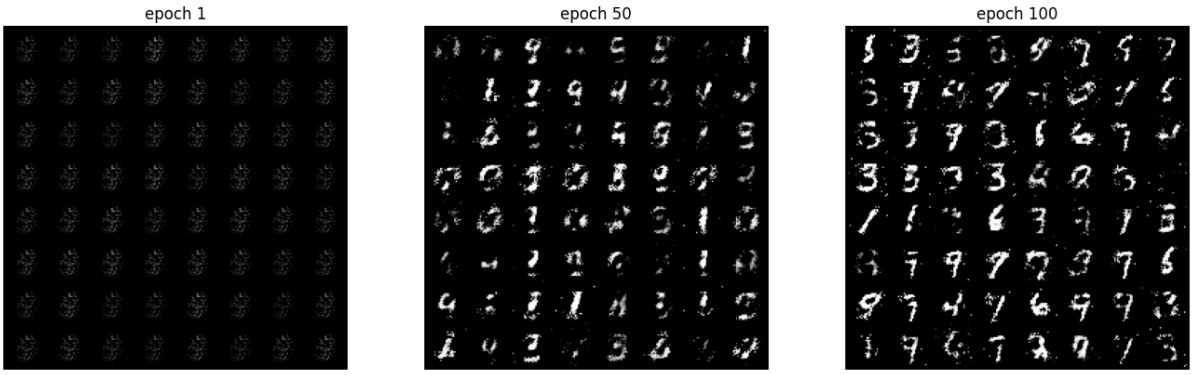


Figure 7: Solution Learning curve



(a) Epoch 1 (b) Epoch 50 (c) Epoch 100

Figure 8: Solution Generated images by G

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)

$$x = \frac{1}{2}(u + v) \text{and} y = \frac{1}{2}(u - v)$$

$$J = \begin{bmatrix} \frac{\frac{1}{2}\partial(u+v)}{\partial u} & \frac{\frac{1}{2}\partial(u+v)}{\partial v} \\ \frac{\frac{1}{2}\partial(u-v)}{\partial u} & \frac{\frac{1}{2}\partial(u-v)}{\partial v} \end{bmatrix}$$

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

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

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

(10 pts)

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

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

Thus we solve for $f_{U,V}(u,v) = \frac{1}{2} f_{X,Y}(\frac{1}{2}(u+v), \frac{1}{2}(u-v))$

$$\begin{aligned} f_{U,V} &= \frac{1}{2}\frac{1}{2\pi} exp(-\frac{1}{4}(u+v)^2 \frac{1}{2}) exp(-\frac{1}{4}(u-v)^2 \frac{1}{2}) \\ &= \frac{1}{4\pi} exp(-\frac{1}{8}(u^2+v^2+2uv)) exp(-\frac{1}{8}(u^2+v^2-2uv)) \\ &= \frac{1}{4\pi} exp(-\frac{u^2}{4}) exp(-\frac{v^2}{4}) \\ \mathbf{f_{U,V}} &= \left(\frac{\mathbf{1}}{\sqrt{\mathbf{2}}\sqrt{\mathbf{2}\pi}} \mathbf{exp}\left(-\frac{\mathbf{u^2}}{\mathbf{4}}\right)\right)\left(\frac{\mathbf{1}}{\sqrt{\mathbf{2}}\sqrt{\mathbf{2}\pi}} \mathbf{exp}\left(-\frac{\mathbf{v^2}}{\mathbf{4}}\right)\right) \end{aligned}$$

(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)

$$|det(J)^{-1}| = 2$$

Thus we solve for $f_{X,Y}(x,y) = 2 f_{U,V}(x+y, x-y)$

$$\begin{aligned} f_{X,Y} &= 2\left(\frac{1}{\sqrt{2}\sqrt{2\pi}} exp\left(-\frac{(x+y)^2}{4}\right)\right)\left(\frac{1}{\sqrt{2}\sqrt{2\pi}} exp\left(-\frac{(x-y)^2}{4}\right)\right) \\ &= 2\frac{1}{4\pi} exp\left(-\frac{(x+y)^2+(x-y)^2}{4}\right) \\ &= \frac{1}{2\pi} exp\left(-\frac{2x^2+2y^2}{4}\right) \\ &= \frac{\mathbf{1}}{\mathbf{2}\pi} \mathbf{exp}\left(-\frac{\mathbf{x^2}}{\mathbf{2}}\right) \mathbf{exp}\left(-\frac{\mathbf{y^2}}{\mathbf{2}}\right) \end{aligned}$$

Hence we see that our original joint distribution was recovered.

3 Directed Graphical Model [20 points]

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

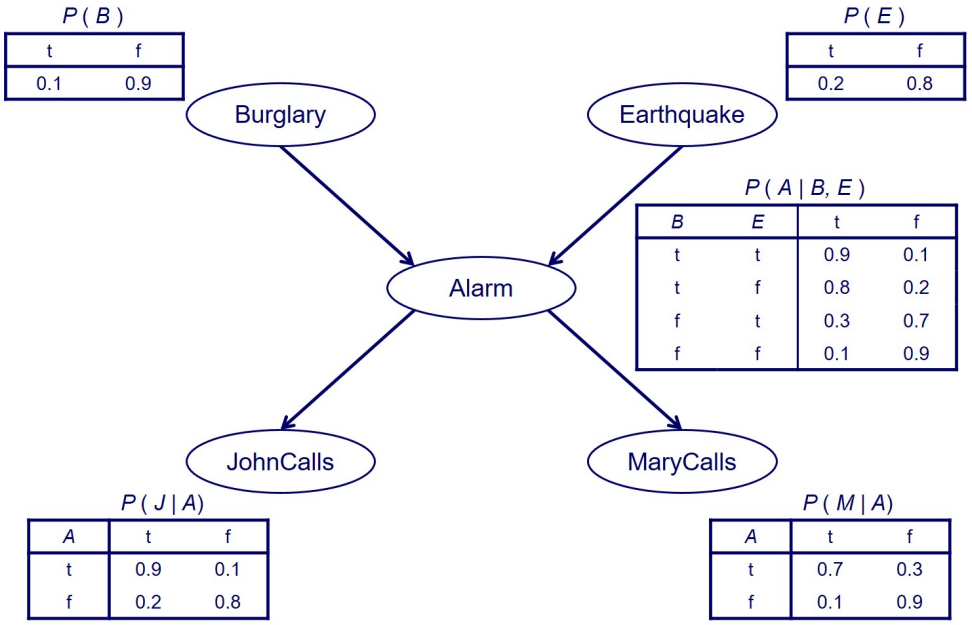


Figure 9: A Bayesian Network example.

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.

Denote short hand notations as B_f meaning $B = f$ or E_t meaning $E = t$. Thus using Bayes Rule we can write:

$$P(B_t \mid E_f, J_t, M_t) = \frac{P(B_t, E_f, J_t, M_t)}{P(B_t, E_f, J_t, M_t) + P(B_f, E_f, J_t, M_t)}$$

Using condition probabilities we can write:

$$P(B_t, E_f, J_t, M_t) = P(B_t)P(E_f)P(A_t \mid B_t, E_f)P(J_t \mid A_t)P(M_t \mid A_t) + P(B_t)P(E_f)P(A_f \mid B_t, E_f)P(J_t \mid A_f)P(M_t \mid A_f)$$

$$P(B_t, E_f, J_t, M_t) = 0.1 * 0.8 * 0.8 * 0.9 * 0.7 + 0.1 * 0.8 * 0.2 * 0.2 * 0.1 = 0.04032 + 0.00032 = 0.04064$$

$$P(B_f, E_f, J_t, M_t) = P(B_f)P(E_f)P(A_t \mid B_f, E_f)P(J_t \mid A_t)P(M_t \mid A_t) + P(B_f)P(E_f)P(A_f \mid B_f, E_f)P(J_t \mid A_f)P(M_t \mid A_f)$$

$$P(B_f, E_f, J_t, M_t) = 0.9 * 0.8 * 0.1 * 0.9 * 0.7 + 0.9 * 0.8 * 0.9 * 0.2 * 0.1 = 0.04536 + 0.01296 = 0.05832$$

$$\begin{aligned} P(B_t \mid E_f, J_t, M_t) &= \frac{P(B_t, E_f, J_t, M_t)}{P(B_t, E_f, J_t, M_t) + P(B_f, E_f, J_t, M_t)} \\ &= \frac{0.04064}{0.04064 + 0.05832} \\ \mathbf{P(B = t \mid E = f, J = t, M = t)} &= \mathbf{0.41067} \end{aligned}$$

Similarily using the same for the second equation:

$$P(B_t \mid E_t, J_t, M_t) = \frac{P(B_t, E_t, J_t, M_t)}{P(B_t, E_t, J_t, M_t) + P(B_f, E_t, J_t, M_t)}$$

Using condition probabilities we can write:

$$P(B_t, E_t, J_t, M_t) = P(B_t)P(E_t)P(A_t \mid B_t, E_t)P(J_t \mid A_t)P(M_t \mid A_t) + P(B_t)P(E_t)P(A_f \mid B_t, E_t)P(J_t \mid A_f)P(M_t \mid A_f)$$

$$P(B_t, E_t, J_t, M_t) = 0.1 * 0.2 * 0.9 * 0.9 * 0.7 + 0.1 * 0.2 * 0.1 * 0.2 * 0.1 = 0.01134 + 0.00004 = 0.01138$$

$$P(B_f, E_t, J_t, M_t) = P(B_f)P(E_t)P(A_t \mid B_f, E_t)P(J_t \mid A_t)P(M_t \mid A_t) + P(B_f)P(E_t)P(A_f \mid B_f, E_t)P(J_t \mid A_f)P(M_t \mid A_f)$$

$$P(B_f, E_t, J_t, M_t) = 0.9 * 0.2 * 0.3 * 0.9 * 0.7 + 0.9 * 0.2 * 0.7 * 0.2 * 0.1 = 0.03402 + 0.00252 = 0.03654$$

$$P(B_t \mid E_t, J_t, M_t) = \frac{P(B_t, E_t, J_t, M_t)}{P(B_t, E_t, J_t, M_t) + P(B_f, E_t, J_t, M_t)} = \frac{0.01138}{0.01138 + 0.03654}$$

$$\mathbf{P(B = t \mid E = t, J = t, M = t) = 0.23748}$$

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

Lets first write the probabilities:

- $P_X(T) = 0.5, P_X(F) = 0.5$
- $P_Y(T) = 0.5, P_Y(F) = 0.5$
- $P_Z(T) = 0.55, P_Z(F) = 0.45$
- $P_{X,Y}(T,T) = 0.4, P_{X,Y}(T,F) = 0.1, P_{X,Y}(F,T) = 0.1, P_{X,Y}(F,F) = 0.4$
- $P_{Y,Z}(T,T) = 0.45, P_{Y,Z}(T,F) = 0.05, P_{Y,Z}(F,T) = 0.1, P_{Y,Z}(F,F) = 0.4$
- $P_{X,Z}(T,T) = 0.38, P_{X,Z}(T,F) = 0.12, P_{X,Z}(F,T) = 0.17, P_{X,Z}(F,F) = 0.33$

1. Compute the mutual information $I(X, Y)$ based on the frequencies observed in the data. (5 pts)

$$I(X;Y) = 0.4 * \log_2(\frac{0.4}{0.5 * 0.5}) + 0.1 * \log_2(\frac{0.1}{0.5 * 0.5}) + 0.1 * \log_2(\frac{0.1}{0.5 * 0.5}) + 0.4 * \log_2(\frac{0.4}{0.5 * 0.5})$$
$$= 0.27122 - 0.1322 - 0.1322 + 0.27122 = \mathbf{0.27807}$$

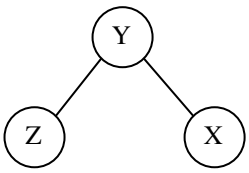
2. Compute the mutual information $I(X, Z)$ based on the frequencies observed in the data. (5 pts)

$$I(X;Z) = 0.38 * \log_2(\frac{0.38}{0.5 * 0.55}) + 0.12 * \log_2(\frac{0.12}{0.5 * 0.45}) + 0.1 * \log_2(\frac{0.17}{0.5 * 0.55}) + 0.4 * \log_2(\frac{0.33}{0.5 * 0.45})$$
$$= 0.17729 - 0.10882 - 0.11796 + 0.18233 = \mathbf{0.13284}$$

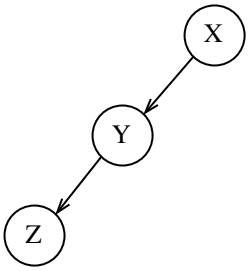
3. Compute the mutual information $I(Z, Y)$ based on the frequencies observed in the data. (5 pts)

$$I(Z;Y) = 0.45 * \log_2(\frac{0.45}{0.5 * 0.55}) + 0.05 * \log_2(\frac{0.05}{0.5 * 0.45}) + 0.1 * \log_2(\frac{0.1}{0.5 * 0.55}) + 0.4 * \log_2(\frac{0.4}{0.5 * 0.45})$$
$$= 0.31972 - 0.10849 - 0.14594 + 0.33202 = \mathbf{0.39731}$$

4. Which undirected edges will be selected by the Chow-Liu algorithm as the maximum spanning tree? (5 pts)
We arrange the Mutual Information in descending order as below:
$I = \{I(Z, Y), I(X, Y), I(X, Z)\}$ According to Chow Liu we first connect the nodes with highest Mutual Information, hence we first connect Y to Z and then Y to X and X to Z are connected already.



5. Root your tree at node X, assign directions to the selected edges. (5 pts)



References

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. *Advances in neural information processing systems*, 27.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.