Assignment #5

Due: 11:59pm EST, April 8, 2021

Homework 5: EM with Mixtures, PCA, and Graphical Models

This homework assignment will have you work with EM for mixtures, PCA, and graphical models. We encourage you to read sections 9.4 and 8.2.5 of the course textbook.

Please type your solutions after the corresponding problems using this LaTeX template, and start each problem on a new page.

Please submit the **writeup PDF to the Gradescope assignment 'HW5'**. Remember to assign pages for each question.

Please submit your **LaTeX file and code files to the Gradescope assignment 'HW5 - Supplemental'**.

Problem 1 (Expectation-Maximization for Gamma Mixture Models, 25pts)

In this problem we will explore expectation-maximization for a Categorical-Gamma Mixture model.

Let us suppose the following generative story for an observation x: first one of K classes is randomly selected, and then the features x are sampled according to this class. If

$$z \sim \text{Categorical}(\boldsymbol{\theta})$$

indicates the selected class, then x is sampled according to the class or "component" distribution corresponding to z. (Here, $\boldsymbol{\theta}$ is the mixing proportion over the K components: $\sum_k \theta_k = 1$ and $\theta_k > 0$). In this problem, we assume these component distributions are gamma distributions with shared shape parameter but different rate parameters:

$$x|z \sim \text{Gamma}(\alpha, \beta_k).$$

In an unsupervised setting, we are only given a set of observables as our training dataset: $\mathcal{D} = \{x_n\}_{n=1}^N$. The EM algorithm allows us to learn the underlying generative process (the parameters $\boldsymbol{\theta}$ and $\{\beta_k\}$) despite not having the latent variables $\{z_n\}$ corresponding to our training data.

1. **Intractability of the Data Likelihood** We are generally interested in finding a set of parameters β_k that maximizes the likelihood of the observed data:

   $$\log p(\{x_n\}_{n=1}^N; \boldsymbol{\theta}, \{\beta_k\}_{k=1}^K).$$

   Expand the data likelihood to include the necessary sums over observations x_n and to marginalize out the latents $\mathbf{z}_n$. Why is optimizing this likelihood directly intractable?

2. **Complete Data Log Likelihood** The complete dataset $\mathcal{D} = \{(x_n, \mathbf{z}_n)\}_{n=1}^N$ includes latents $\mathbf{z}_n$. Write out the negative complete data log likelihood:

   $$\mathcal{L}(\boldsymbol{\theta}, \{\beta_k\}_{k=1}^K) = -\log p(\mathcal{D}; \boldsymbol{\theta}, \{\beta_k\}_{k=1}^K).$$

   Apply the power trick and simplify your expression using indicator elements z_{nk}.[a] Notice that optimizing this loss is now computationally tractable if we know $\mathbf{z}_n$.

(Continued on next page.)

[a] The "power trick" is used when terms in a PDF are raised to the power of indicator components of a one-hot vector. For example, it allows us to rewrite $p(\mathbf{z}_n; \boldsymbol{\theta}) = \prod_k \theta_k^{z_{nk}}$.

Problem 1 (cont.)

3. **Expectation Step** Our next step is to introduce a mathematical expression for $\mathbf{q}_n$, the posterior over the hidden component variables $\mathbf{z}_n$ conditioned on the observed data x_n with fixed parameters. That is:

$$\mathbf{q}_n = \begin{bmatrix} p(\mathbf{z}_n = \mathbf{C}_1 | x_n; \boldsymbol{\theta}, \{\beta_k\}_{k=1}^K) \\ \vdots \\ p(\mathbf{z}_n = \mathbf{C}_K | x_n; \boldsymbol{\theta}, \{\beta_k\}_{k=1}^K) \end{bmatrix}.$$

   Write down and simplify the expression for $\mathbf{q}_n$. Note that because the $\mathbf{q}_n$ represents the posterior over the hidden categorical variables $\mathbf{z}_n$, the components of vector $\mathbf{q}_n$ must sum to 1. The main work is to find an expression for $p(\mathbf{z}_n | x_n; \boldsymbol{\theta}, \{\beta_k\}_{k=1}^K)$ for any choice of $\mathbf{z}_n$; i.e., for any 1-hot encoded $\mathbf{z}_n$. With this, you can then construct the different components that make up the vector $\mathbf{q}_n$.

4. **Maximization Step** Using the $\mathbf{q}_n$ estimates from the Expectation Step, derive an update for maximizing the expected complete data log likelihood in terms of $\boldsymbol{\theta}$ and $\{\beta_k\}_{k=1}^K$.
   (a) Derive an expression for the expected complete data log likelihood using $\mathbf{q}_n$.
   (b) Find an expression for $\boldsymbol{\theta}$ that maximizes this expected complete data log likelihood. You may find it helpful to use Lagrange multipliers in order to enforce the constraint $\sum \theta_k = 1$. Why does this optimal $\boldsymbol{\theta}$ make intuitive sense?
   (c) Find an expression for β_k that maximizes the expected complete data log likelihood. Why does this optimal β_k make intuitive sense?

5. Suppose that this had been a classification problem. That is, you were provided the "true" components $\mathbf{z}_n$ for each observation x_n, and you were going to perform the classification by inverting the provided generative model (i.e. now you're predicting $\mathbf{z}_n$ given x_n). Could you reuse any of your derivations above to estimate the parameters of the model?

6. Finally, implement your solution in `p1.ipynb` and attach the final plot below.

   You will receive no points for code not included below.

Solution

1. The log-likelihood we are trying to maximize is

$$\log p(\{x_n\}_{n=1}^N; \boldsymbol{\theta}, \{\beta_k\}_{k=1}^K)$$
$$= \sum_n \log p(x_n; \boldsymbol{\theta}\{\beta_k\}_{k=1}^K)$$

   marginalizing out our latent $\mathbf{z}_n$'s we have

$$= \sum_n \log \left[\sum_k p(x_n, z_{n,k}; \boldsymbol{\theta}, \beta_k) \right]$$

   we can simplify no further as consolidating a summation inside of a logarithm is not possible; optimizing this likelihood directly is intractable.

2. The complete negative data log likelihood is

$$\begin{aligned}
&\mathcal{L}(\boldsymbol{\theta}, \{\beta_k\}_{k=1}^K) \\
&= -\log p(D; \boldsymbol{\theta}, \{\beta_k\}_{k=1}^K) \\
&= -\log p(\{(x_n, \mathbf{z}_n)\}_{k=1}^K; \boldsymbol{\theta}, \{\beta_k\}_{k=1}^K) \\
&= -\sum_n \log p(x_n, \mathbf{z}_n; \boldsymbol{\theta}, \{\beta_k\}_{k=1}^K) \\
&= -\sum_n \log \left[p(x_n; \mathbf{z}_n, \{\beta_k\}_{k=1}^K) p(\mathbf{z}_n; \boldsymbol{\theta})\right] \\
&= -\sum_n \log p(x_n; \mathbf{z}_n, \{\beta_k\}_{k=1}^K) + \log p(\mathbf{z}_n; \boldsymbol{\theta})
\end{aligned}$$

using the power trick this turns into

$$= -\sum_n \log \left[\prod_k \text{Gamma}(x_n; \alpha, \beta_k)^{z_{n,k}}\right] + \log \left[\prod_k \theta_k^{z_{n,k}}\right]$$

$$\boxed{\mathcal{L}(\boldsymbol{\theta}, \{\beta_k\}_{k=1}^K) = -\sum_n \sum_k z_{n,k} \log \text{Gamma}(x_n; \alpha, \beta_k) + z_{n,k} \log \theta_k}$$

3. Each of our $q_{n,k}$ is

$$q_{n,k} = p(\mathbf{z}_n = C_k | x_n, \boldsymbol{\theta}, \{\beta_k\}_{k=1}^K)$$

by Bayes' rule we have

$$\begin{aligned}
q_{n,k} &\propto p(x_n | \mathbf{z}_n = C_k, \{\beta_k\}_{k=1}^K) p(\mathbf{z}_n = C_k; \boldsymbol{\theta}) \\
&= \text{Gamma}(x_n; \alpha, \beta_k)^{z_{n,k}} \theta_k
\end{aligned}$$

and to get the entire vector $\mathbf{q}_n$, we must **normalize these values** such that they sum to one, i.e.

$$\boxed{q_{n,k} = \frac{\text{Gamma}(x_n; \alpha, \beta_k)^{z_{n,k}} \theta_k}{\sum_k \text{Gamma}(x_n; \alpha, \beta_k)^{z_{n,k}} \theta_k}}$$

4. (a) What we want to find is the following:

$$\text{E}_{\mathbf{z}_n | x_n} \left[-\mathcal{L}(\boldsymbol{\theta}, \{\beta_k\}_{k=1}^K)\right]$$

we have that this is equal to

$$\begin{aligned}
&= \text{E}_{\mathbf{z}_n | x_n} \left[\sum_n \log p(x_n; \mathbf{z}_n, \{\beta_k\}_{k=1}^K) + \log p(\mathbf{z}_n; \boldsymbol{\theta})\right] \\
&= \sum_n \text{E}_{\mathbf{z}_n | x_n} \left[\log p(x_n; \mathbf{z}_n, \{\beta_k\}_{k=1}^K) + \log p(\mathbf{z}_n; \boldsymbol{\theta})\right] \\
&= \sum_n \sum_k p(\mathbf{z}_n = C_k | x_n) \left(\log p(x_n; \mathbf{z}_n = C_k, \beta_k) + \log p(\mathbf{z}_n = C_k; \boldsymbol{\theta})\right) \\
&= \sum_n \sum_k q_{n,k} \left(\log p(x_n; \mathbf{z}_n = C_k, \beta_k) + \log p(\mathbf{z}_n = C_k; \boldsymbol{\theta})\right)
\end{aligned}$$

$$\boxed{\text{E}_{\mathbf{z}_n | x_n} \left[-\mathcal{L}(\boldsymbol{\theta}, \{\beta_k\}_{k=1}^K)\right] = \sum_n \sum_k q_{n,k} \left(\log \text{Gamma}(x_n; \alpha, \beta_k) + \log \theta_k\right)}$$

(b) We wish to solve for the following

$$\underset{\theta}{\operatorname{argmin}}\, \mathrm{E}_{\mathbf{z}_n|x_n} \mathcal{L}(\boldsymbol{\theta}, \{\beta_k\}_{k=1}^K) \quad \text{s.t.} \quad \sum_k \theta_k - 1 = 0$$

we can do so using a Lagrange multiplier:

$$\begin{aligned}
\mathrm{L}(\boldsymbol{\theta}, \lambda) &= \mathrm{E}_{\mathbf{z}_n|x_n} \mathcal{L}(\boldsymbol{\theta}, \{\beta_k\}_{k=1}^K) + \lambda \left(\sum_{j=1}^K \theta_j - 1 \right) \\
&= -\sum_{i=1}^N \sum_{j=1}^K q_{i,j} \left(\log \mathrm{Gamma}(x_i; \alpha, \beta_j) + \log \theta_j \right) + \lambda \left(\sum_{j=1}^K \theta_j - 1 \right) \\
&= -\sum_{i=1}^N \sum_{j=1}^K q_{i,j} \left(\alpha \log \beta_j - \log \Gamma(x_i) + (\alpha - 1) \log(x_i) - \beta_j x + \log \theta_j \right) + \lambda \left(\sum_{j=1}^K \theta_j - 1 \right)
\end{aligned}$$

the gradient with respect to θ_k of the above is

$$\begin{aligned}
\nabla_{\theta_k} \mathrm{L}(\boldsymbol{\theta}, \lambda) &= -\sum_{i=1}^N \nabla_{\theta_k} q_{i,k} (\log \theta_k) + \lambda \\
&= -\sum_{i=1}^N \frac{q_{i,k}}{\theta_k} + \lambda \\
&= -\frac{N}{\theta_k} \sum_{i=1}^N q_{i,k} + \lambda
\end{aligned}$$

and setting this to zero to solve for θ_k yields

$$\boxed{\theta_k = -\frac{N}{\lambda} \sum_{i=1}^N q_{i,k}}$$

now, we do the same procedure to solve for λ by taking the following gradient and setting it to zero:

$$\nabla_\lambda \mathrm{L}(\boldsymbol{\theta}, \lambda) = \sum_{j=1}^K \theta_j - 1$$

substituting the boxed equation from above we have

$$\begin{aligned}
\nabla_\lambda \mathrm{L}(\boldsymbol{\theta}, \lambda) &= \sum_{j=1}^K -\frac{N}{\lambda} \sum_{i=1}^N q_{i,j} - 1 \\
&= -\frac{N}{\lambda} \sum_{j=1}^K \sum_{i=1}^N q_{i,j} - 1
\end{aligned}$$

notice that $\sum_{j=1}^{K}\sum_{i=1}^{N} q_{i,j} = N$ so we have, setting to zero,

$$\nabla_\lambda \mathrm{L}(\boldsymbol{\theta}, \lambda) = -\frac{N^2}{\lambda} - 1 = 0$$
$$\implies \boxed{\lambda = -N^2}$$

putting together our two boxed equations we have

$$\theta_k = -\frac{N}{-N^2}\sum_{i=1}^{N} q_{i,k}$$
$$\boxed{\theta_k = \frac{1}{N}\sum_{i=1}^{N} q_{i,k}}$$

this optimal value for θ_k makes sense we are averaging, over the entire data set, the entry of $\mathbf{q_n}$ that corresponds to the category C_k.

(c) We wish to solve for the following

$$\underset{\{\beta_k\}_{k=1}^{K}}{\operatorname{argmin}} \mathrm{E}_{\mathbf{z}_n|x_n}\mathcal{L}(\boldsymbol{\theta}, \{\beta_k\}_{k=1}^{K})$$

we can just differentiate with respect to β_k here to solve this:

$$\nabla_{\beta_k}\mathrm{E}_{\mathbf{z}_n|x_n}\mathcal{L}(\boldsymbol{\theta}, \{\beta_k\}_{k=1}^{K})$$
$$= \nabla_{\beta_k} - \sum_{i=1}^{N}\sum_{j=1}^{K} q_{i,j}\left(\log \mathrm{Gamma}(x_i; \alpha, \beta_j) + \log\theta_j\right)$$
$$= \nabla_{\beta_k} - \sum_{i=1}^{N}\sum_{j=1}^{K} q_{i,j}\left(\alpha\log\beta_j - \log\Gamma(x_i) + (\alpha-1)\log(x_i) - \beta_j x + \log\theta_j\right)$$
$$= -\sum_{i=1}^{N}\nabla_{\beta_k}[q_{i,k}\alpha\log\beta_k - q_{i,k}\beta_k x_i]$$
$$= -\sum_{i=1}^{N}\frac{q_{i,k}\alpha}{\beta_k} - q_{i,k}x_i$$
$$= -\sum_{i=1}^{N}\frac{q_{i,k}\alpha}{\beta_k} - q_{i,k}x_i$$

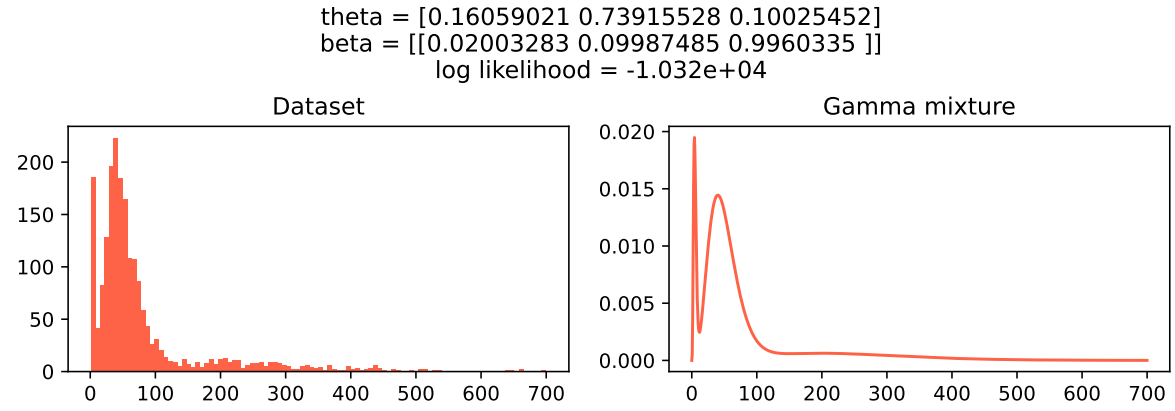
setting this to zero yields

$$\boxed{\beta_k = \alpha\frac{\sum_{i=1}^{N} q_{i,k}}{\sum_{i=1}^{N} q_{i,k}x_i}}$$

this answer makes intuitive sense as β_k, the rate parameter for the gamma distribution, is being calculated to be negatively proportional to a weighted average of the x_n's, which could be interpreted as times between events (such that their average is the inverse of the rate).

5. Yes, except that we would no longer have to average over a "soft" definition of $\mathbf{z}_n$, which is how we used $\mathbf{q}_n$. When calculating our loss function to minimize, instead we could directly plug in the one-hot encoded vector $\mathbf{z}_n$

6. Plot:



Code:

```
def e_step(theta, betas):
    scales = 1 / betas
    x_given_z = gamma(alpha, scale=scales)
    q = np.multiply(x_given_z.pdf(x), theta)
    s = np.sum(q, axis=1)
    q_normalized = np.array([q[i] / s[i] for i in range(len(q))])
    return q_normalized


def m_step(q):
    theta_hat = np.sum(q, 0) / q.shape[0]
    beta_hats = alpha * np.sum(q, 0) / (x.T @ q)
    return theta_hat, beta_hats


def log_px(x, theta, betas):
    scales = 1 / betas
    x_given_z = gamma(alpha, scale=scales)
    log_px = np.log(np.sum(np.exp(x_given_z.logpdf(x) + np.log(theta)), axis=1))
    return log_px


def run_em(theta, betas, iterations=1000):
    for _ in range(iterations):
        q = e_step(theta, betas)
        theta, betas = m_step(q)
    return theta, betas
```

Problem 2 (PCA, 15 pts)

For this problem you will implement PCA from scratch on the first 6000 images of the MNIST dataset. Your job is to apply PCA on MNIST and discuss what kind of structure is found. Implement your solution in `p2.ipynb` and attach the final plots below.

You will recieve no points for using third-party PCA implementations (i.e. `scikit-learn`).

You will recieve no points for code not included below.

1. Compute the PCA. Plot the eigenvalues corresponding to the most significant 500 components in order from most significant to least. Make another plot that describes the cumulative proportion of variance explained by the first k most significant components for values of k from 1 through 500. How much variance is explained by the first 500 components? Describe how the cumulative proportion of variance explained changes with k. Include this plot below.

2. Plot the mean image of the dataset and plot an image corresponding to each of the first 10 principle components. How do the principle component images compare to the cluster centers from K-means? Discuss any similarities and differences. Include these two plots below.

   Reminder: Center the data before performing PCA

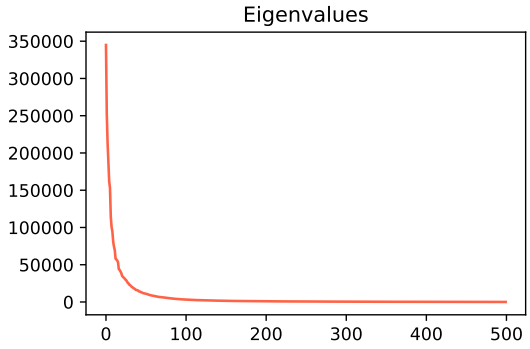
3. Compute the reconstruction error on the data set using the mean image of the dataset. Then compute the reconstruction error using the first 10 principal components. How do these errors compare to the final objective loss achieved by using K-means on the dataset? Discuss any similarities and differences.

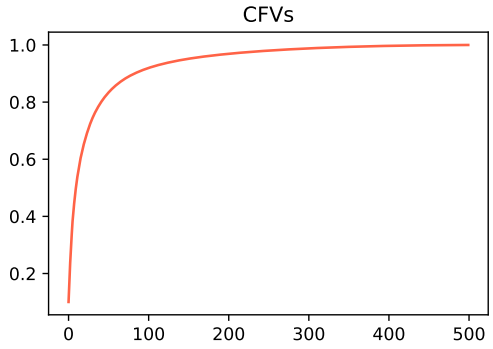
   For consistency in grading, define the reconstruction error as the squared L2 norm averaged over all data points.

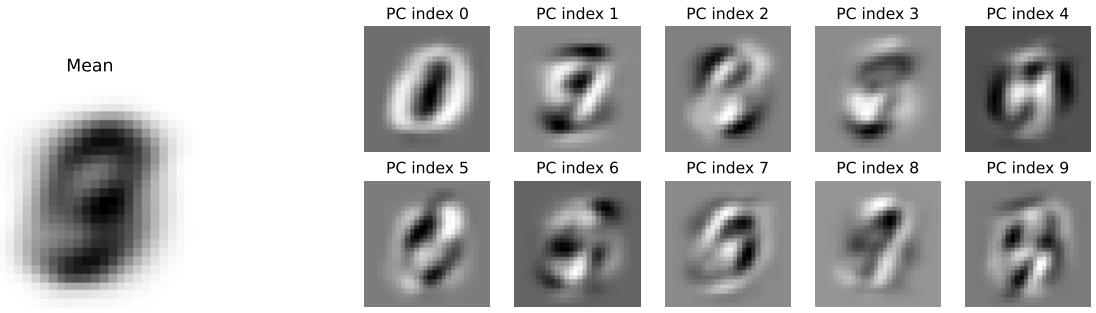
4. Suppose you took the original matrix of principle components that you found U and multiplied it by some rotation matrix R. Would that change the quality of the reconstruction error in the last problem? The interpretation of the components? Why or why not?

Solution

Plots:







Code:

```
def pca(x, n_comps=500):
    x = x - np.mean(x, axis=0)
    _, s, vh = np.linalg.svd(x)
    top_eigvals = s[:n_comps] ** 2 / x.shape[0]
    top_pcomps = vh[:n_comps, :]
    return top_eigvals, top_pcomps


def calc_cfvs(eigvals):
    cum_frac_vars = np.cumsum(eigvals, axis=0) / np.sum(eigvals)
    return cum_frac_vars


def calc_errs(x, pcomps):
    x_mean = np.mean(x, axis=0)
    err_mean =np.mean(np.sum((x - x_mean) ** 2, axis=1), axis=0)
    err_pcomp = np.mean(np.sum((x - (x_mean + (x - x_mean) @ pcomps[:10, :].T @ pcomps[:10,
                                                    :] )) ** 2, axis=1))
    return err_mean, err_pcomp
```

1. The first 500 eigenvalues of the decomposition correspond to 0.9994 of the variance. This value was calculated by running $PCA(x, n\_comps = 500)$ as well as $PCA(x, n\_comps = 784)$ and calculating the sum of the former divided by the sum of the latter.
2. I did not notice much similarity to the centers from k-means. I think this is a correct evaluation, as the two methods are doing something entirely different; k-means is averaging images it decides are most

similar, while PCA is selecting features that explain the most variance. Because of this, it makes sense that k-means may somewhat resemble actual digits whereas the principle components don't resemble anything like digits.

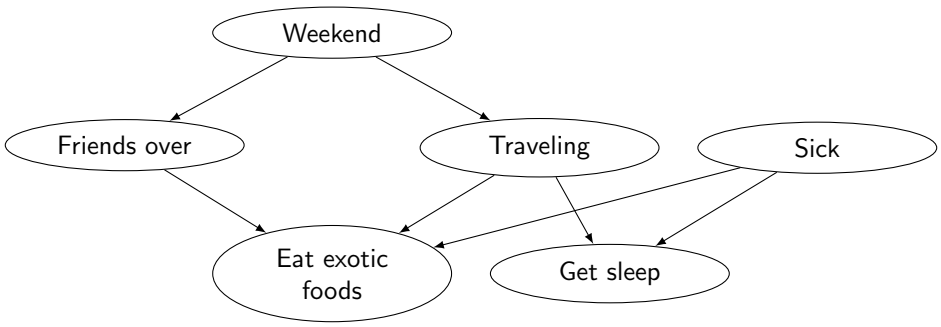
3. For PCA, the reconstruction error using the mean is 3.436024e+06, and the reconstruction error using the 10 principle components is 1.731315e+06. The final objective loss achieved by k-means on the dataset was 2.548403e+06. It makes sense that the reconstruction error using the mean image is the highest as it is essentially calculating the k-means objective error on one cluster. The reasoning behind the reconstruction error using the 10 principle components being lower than the k-means objective error is more subtle–it is because k-means is simply grouping close images together whereas PCA is purposefully selecting images that together explain the most variance in the data set.

4. This would not affect either the quality of the reconstruction error as we are changing the basis of the principle components which affects both dimensionality reduction and reconstruction equivalently, meaning it gets cancelled out in the reconstruction error calculation.

   The interpretation of the components would change, however, as these would be completely changed relative to the data set whose variance they are trying to maximally represent.

Problem 3 (Bayesian Networks, 10 pts)

In this problem we explore the conditional independence properties of a Bayesian Network. Consider the following Bayesian network representing a fictitious person's activities. Each random variable is binary (true/false).



The random variables are:

- Weekend: Is it the weekend?
- Friends over: Does the person have friends over?
- Traveling: Is the person traveling?
- Sick: Is the person sick?
- Eat exotic foods: Is the person eating exotic foods?
- Get Sleep: Is the person getting sleep?

For the following questions, $A \perp B$ means that events A and B are independent and $A \perp B|C$ means that events A and B are independent conditioned on C.

Use the concept of d-separation to answer the questions and show your work (i.e., state what the blocking path(s) is/are and what nodes block the path; or explain why each path is not blocked).

Example Question: Is Friends over $\perp$ Traveling? If NO, give intuition for why.

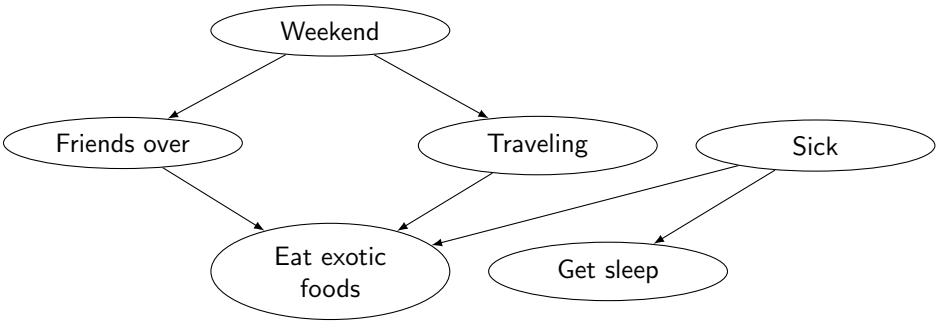
Example Answer: NO. The path from Friends over – Weekend – Traveling is not blocked following the d-separation rules as we do not observe Weekend. Thus, the two are not independent.

Actual Questions:

1. Is Weekend $\perp$ Get Sleep? If NO, give intuition for why.
2. Is Sick $\perp$ Weekend? If NO, give intuition for why.
3. Is Sick $\perp$ Friends over | Eat exotic foods? If NO, give intuition for why.
4. Is Friends over $\perp$ Get Sleep? If NO, give intuition for why.
5. Is Friends over $\perp$ Get Sleep | Traveling? If NO, give intuition for why.
6. Suppose the person stops traveling in ways that affect their sleep patterns. Travel still affects whether they eat exotic foods. Draw the modified network. (Feel free to reference the handout file for the commands for displaying the new network in LaTeX).
7. For this modified network, is Friends over $\perp$ Get Sleep? If NO, give an intuition why. If YES, describe what observations (if any) would cause them to no longer be independent.

Solution

1. No, if traveling is not observed. If traveling is not observed than the information flow is unblocked.
2. Yes. There is no information flow between the two. This is assuming get sleep is not observed.
3. No. Explaining away is occurring. If we observe eat exotic foods, then knowing sick can tell us how much friends over contributed to eat exotic foods, and vice versa.
4. No. Unless weekend, is observed, than knowing friends over gives information on weekend, which in turn gives information on get sleep.
5. Yes. Once weekend is observed then knowing friends over gives zero information on get sleep.
6. The new network is the following



7. Yes. If eat exotic foods was observe, than there would be explaining away occurring between friends over and sick such that knowing if friends over would give information on sick. This information would in turn give information on get sleep, so there would be information flow rendering friends over and get sleep not independent.

Name

Rodney Lafuente

Collaborators and Resources

Whom did you work with, and did you use any resources beyond cs181-textbook and your notes? I did not work with anyone on this problem set.

Calibration

Approximately how long did this homework take you to complete (in hours)? 20+