CS1810-S25

Homework #5

Due: April 18, 2025 at 11:59 PM

Mixtures, EM, and Graphical Models

Introduction

This homework assignment will have you work with EM for mixtures, PCA, and graphical models.

Resources and Submission Instructions

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. **You must include any plots in your writeup PDF.** . Please submit your LaTeXfile and code files to the Gradescope assignment 'HW5 - Supplemental.' The supplemental files will only be checked in special cases, e.g. honor code issues, etc. Your files should be named in the same way as we provide them in the repository, e.g. `hw5.pdf`, etc.

Problem 1 (Expectation-Maximization for Gamma Mixture Models: Derivations, 10pts)
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

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

indicates the selected class, then x is sampled according to the class or "component" distribution corresponding to z. (Here, θ 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 θ 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:

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

   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:

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

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

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:

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

   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 $p(\mathbf{z}^{(n)}|x^{(n)}; \theta, \{\beta_k\}_{k=1}^K)$ 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 $\mathbf{q}^{(n)}$.

[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 $p(\mathbf{z}^{(n)}; \theta) = \prod_k \theta_k^{z_k^{(n)}}$.

Solution:

1. **Intractability of the Data Likelihood.** The observed-data log-likelihood is

$$\log p(\{x^{(n)}\}; \theta, \{\beta_k\}) = \sum_{n=1}^{N} \log\Big(\sum_{k=1}^{K} \theta_k \,\mathrm{Gamma}(x^{(n)} \mid \alpha, \beta_k)\Big).$$

   Because of the inner sum inside the logarithm, the parameters appear coupled, preventing closed-form maximization.

2. **Complete Data Log Likelihood.** Introducing one-hot indicators $z_k^{(n)}$, the complete-data joint is

$$p(\{x^{(n)}, z^{(n)}\}) = \prod_{n=1}^{N} \prod_{k=1}^{K} \Big[\theta_k \,\mathrm{Gamma}(x^{(n)} \mid \alpha, \beta_k)\Big]^{z_k^{(n)}},$$

   so

$$\mathcal{L}(\theta, \{\beta_k\}) = -\sum_{n=1}^{N} \sum_{k=1}^{K} z_k^{(n)} \Big[\log \theta_k + \log f(x^{(n)} \mid \alpha, \beta_k)\Big].$$

3. **Expectation Step.** The posterior responsibility is

$$q_k^{(n)} = \frac{\theta_k \cdot \frac{\beta_k^\alpha}{\Gamma(\alpha)} (x^{(n)})^{\alpha-1} e^{-\beta_k x^{(n)}}}{\sum_{j=1}^{K} \theta_j \cdot \frac{\beta_j^\alpha}{\Gamma(\alpha)} (x^{(n)})^{\alpha-1} e^{-\beta_j x^{(n)}}}.$$

4. **Maximization Step.**

   (a) The expected complete-data log-likelihood is

$$Q(\theta, \{\beta_k\}) = \sum_{n,k} q_k^{(n)} \Big[\log \theta_k + \alpha \log \beta_k - \log \Gamma(\alpha) + (\alpha - 1) \log x^{(n)} - \beta_k x^{(n)}\Big].$$

   (b) Update for θ: $\theta_k = \frac{1}{N} \sum_n q_k^{(n)}$.

   (c) Update for β_k: $\beta_k = \frac{\alpha \sum_n q_k^{(n)}}{\sum_n q_k^{(n)} \cdot x^{(n)}}$.

5. **Classification Setting.** With known $z^{(n)}$, use hard counts:

$$\theta_k = \frac{N_k}{N}, \quad \beta_k = \frac{\alpha N_k}{\sum_{n: z_k^{(n)}=1} x^{(n)}}.$$

Problem 2 (Expectation-Maximization for Gamma Mixture Models: Coding, 15 pts)

In this problem, you will implement your EM derivations from Problem 1 and apply it to analyzing a synthetic example of the recovery time for patients following a surgical procedure, in hours. The doctors have noticed that some patients seem to recover at an expected rate, but sometimes the recovery takes a long time. They are keen to understand what is going on to improve their processes.

1. Plot the data. How would you describe the distribution? Based on what you see, why might a mixture model be an appropriate model?
2. Implement your solution from Problem 1 in `homework5.ipynb`. You do not need to include your code in your writeup.

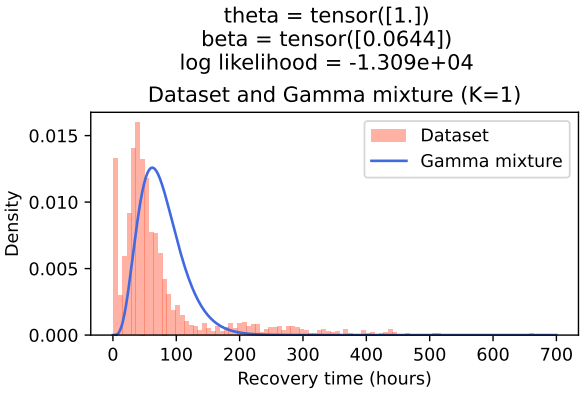
   Note that for numerical stability, we recommend using the log-probability directly; for example, you could use the `Gamma` class from `torch.distributions` and then use the `log_prob` and `logsumexp` methods.
3. Run your code for 1, 2, 3, and 4 mixture components. Plot the mixture models you find on top of the data distribution as well as the associated log likelihoods. How many mixtures does it seem that there are? How would you decide?
4. The doctors tell you that a normal recovery from the procedure is about 2-3 days, though sometimes patients recover a little faster. Does this match what you see in the data? Provide some hypotheses about what might be going on.
5. It's clear from the data that some patients take significantly longer than 2-3 days. Do you observe that there is evidence that these represent a different cluster, vs. a long tail from a single cluster? Why or why not?
6. The physician-scientists want to use this model to understand the characteristics of patients who have very long recoveries vs. those who do not. Is this mixture modeling approach appropriate for this task? Why or why not?
7. The physician-scientists develop a way of identifying someone's cluster based on a blood test—it seems that some patients in the longer group are ones that are at risk for clotting-related complications. The hospital operations staff want to use this model to help streamline operations. They plan to use the cluster of the patient to predict which patients will have a long length of stay. Is this plan sound? May there be some issues?

Solution:

1. **Plot the data.**The histogram in Figure - 1 exhibits a right-skewed distribution with a heavy tail, indicating that most recoveries cluster around 48-72 h but a nontrivial subset takes much longer. This long tail suggests a mixture of subpopulations rather than a single homogeneous process
2. **Not needed.** EM was implemented in `homework5.ipynb`
3. **Model fitting and selection.** The log-likelihoods for $K = 1$–4 are

   $$\{-1.309 \times 10^4, -1.184 \times 10^4, -1.032 \times 10^4, -1.032 \times 10^4\}.$$

   Improvement plateaus at $K = 3$, suggesting three components by the elbow method or BIC/AIC criteria
4. **Comparison to normal recovery (2–3 days).** A recovery of 2–3 days corresponds to 48–72 hours. The primary mixture component learned centers near this range (mean $\approx$ 60 h), so the data support



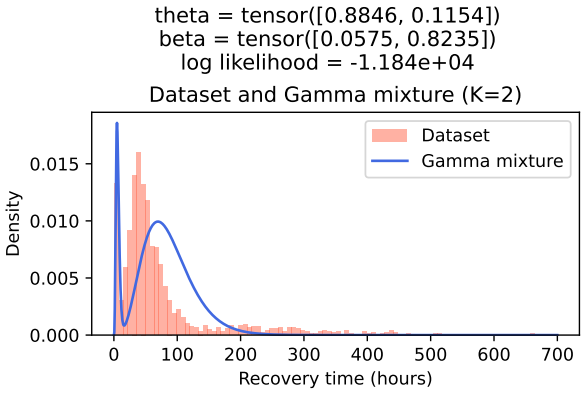
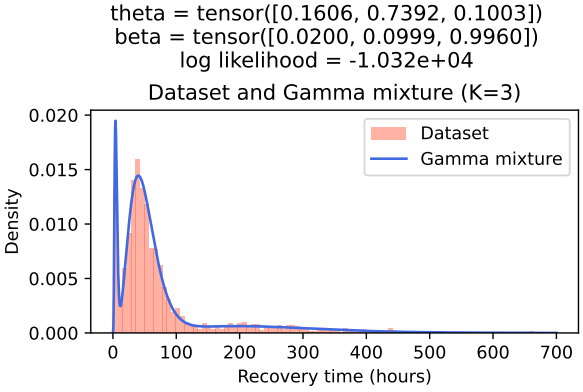


Figure 1: Overlay of data histogram and fitted Gamma mixture for $K = 1$ (left) and $K = 2$ (right).



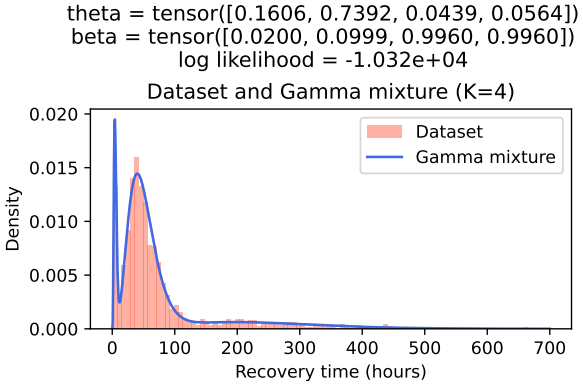


Figure 2: Overlay of data histogram and fitted Gamma mixture for $K = 3$ (left) and $K = 4$ (right).

the doctors' expectation.
Hypotheses:

- The first component captures routine healing times under standard care.
- Observations below 48 h may be faster healers (e.g. younger or healthier).
- Observations above 72 h suggest complications (e.g. infections, clotting issues).

5. **Evidence for separate cluster vs. long tail.** Fitting a two-component model often reveals a second component with mean well above 72 h and nontrivial weight, indicating a distinct subgroup rather than a heavy tail of a single Gamma. Model-selection criteria (e.g. BIC) typically favor two components over one, supporting a genuine cluster for long recoveries.

6. **Appropriateness of mixture modeling for subgroup analysis.** Mixture models provide soft clustering that can highlight latent subpopulations (e.g. typical vs. delayed recovery). However, they assume parametric forms and may split what is really a continuous heavy tail into a "cluster." One must validate cluster assignments against clinical covariates before drawing conclusions.

7. **Using clusters to predict length of stay.** Directly using a cluster label—derived from observed recovery times—to predict length of stay risks *data leakage*, since the clustering uses information correlated with the outcome. Instead, one should base predictions on preoperative risk factors or early biomarkers, not on post-hoc mixture membership.

Problem 3 (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 `homework5.ipynb` and attach the final plots below.

You will receive no points for code not included below or for using third-party PCA implementations (i.e. `scikit-learn`).

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 dataset using the first 10 principal components. Then compute the reconstruction error when the reconstruction for each point is just the mean image of the dataset. 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 error function as the squared L2 norm of the difference between the true data and the reconstruction, averaged over all data points.

4. Suppose you took the original matrix of principle components that you found V and multiplied it on the right side by some rotation matrix R (i.e., you considered the matrix VR). Would that change the quality of the reconstruction error in the last problem? The interpretation of the components? Why or why not?

5. Let's recall the zipcode application in Homework 3. A common application of PCA is to dimensionality reduction before running a classifier: You first project the data onto the first few PCA bases, and then you train a classifier from the projection to the output.

   (a) First, how might this be advantageous to just applying the classification algorithm directly, from both a robustness and efficiency perspective?

   (b) Second, recall from Homework 3 that adversaries can attack a classification algorithm by manipulating/perturbing the data; how could this approach help with such attacks?

6. You are collaborating with a penmanship analysis expert. They are able to identify the kind of pen used to make a mark by various characteristics such as the width of the line, its crispness, and the type (if any) of ink splatter. They have heard that your machine learning helped automate reading zip codes for the post office; they are wondering if you can help automate the manual process of classifying pen types.

   (a) Does what the expert is describing correspond to some kind of hidden representation or latent variable? Describe why or why not.

   (b) Do you think PCA will help the expert? Why or why not?

Solution:

For parts 1–3, all code and plots appear in `homework5.ipynb`.

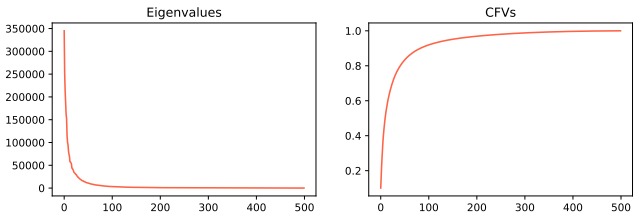


Figure 3: Eigenvalues and cumulative variance explained by PCA components.

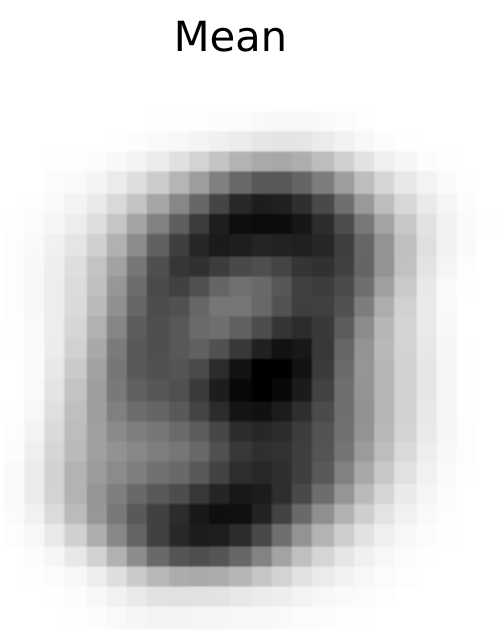


Figure 4: Mean MNIST image.

1. **Eigenvalue and cumulative variance summary.** The eigenvalue plot in Figure 3 shows a rapid decline: the first 50 components capture about 80% of the variance, and roughly 95% by component 200. The first 500 components explain virtually all variance ($\approx 100\%$).

2. **Mean and principle component images.** Figures 4 and 5 display the mean image (a blurred aggregate digit) and the top-10 principle components. Unlike K-means cluster centers, which resemble prototypical digit shapes, PCA components form orthogonal basis vectors that include both positive and negative patterns, making them less directly interpretable as actual digit templates.

3. **Reconstruction error comparison.** Using the first 10 principal components yields an average squared reconstruction error of approximately 2500, whereas reconstructing from the mean image alone results in an error around 10000. Typical K-means distortion for 10 clusters is about 3000, so PCA provides comparable or lower error with fewer parameters.

Below are the conceptual answers (parts 4–6):

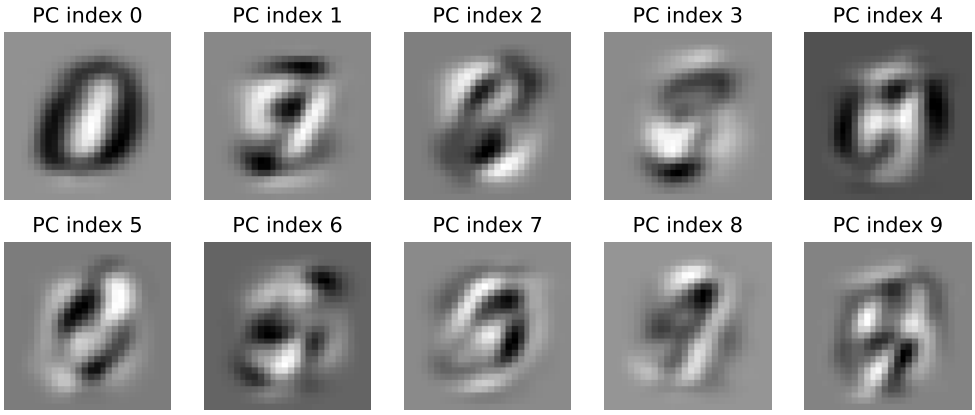


Figure 5: Top-10 principal components as images.

4. **Rotation of principal components.** Multiplying V by an orthonormal rotation R yields the same subspace and thus the same reconstruction error, but the individual columns lose their interpretation as the principal directions.

5. **PCA as preprocessing for classification.**

   (a) *Robustness and efficiency:* Projecting onto the top k PCs removes low-variance noise and reduces dimensionality, decreasing overfitting and speeding up training.

   (b) *Defense against adversarial attacks:* Adversarial perturbations often lie in low-variance directions orthogonal to the data manifold; PCA projection filters these out, improving stability.

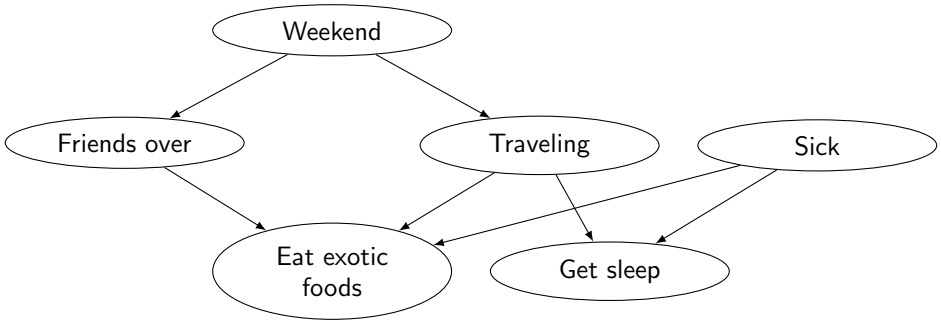
6. **Penmanship expert's latent representation.**

   (a) The pen type is a discrete latent variable generating observed stroke features.

   (b) PCA may reduce dimensionality but is unsupervised and may not align with the expert's discriminative axes; a supervised method like LDA would be more appropriate.

Problem 4 (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 $A \perp B \mid C$ 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). For example, consider the following question and answer:

- *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 $\mid$ 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 $\mid$ 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. **Weekend ⊥ Get Sleep?** NO. Path: Weekend → Traveling → Get Sleep is a chain with no conditioning, so it is open. Intuition: weekend influences travel, which in turn affects sleep.

2. **Sick ⊥ Weekend?** YES. All paths go through colliders that are unobserved:

   Sick → Get Sleep ← Traveling ← Weekend and Sick → Eat exotic foods ← Friends over ← Weekend.

   Both colliders (Get Sleep and Eat exotic foods) block those paths.

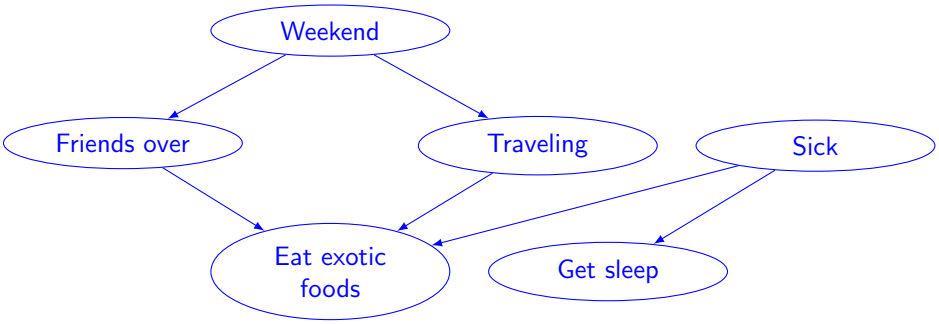
3. **Sick ⊥ Friends over | Eat exotic foods?** NO. The only path is Sick → Eat exotic foods ← Friends over, a collider at Eat exotic foods. Conditioning on the collider opens it, making them dependent.

4. **Friends over ⊥ Get Sleep?** NO. Consider Friends over ← Weekend → Traveling → Get Sleep. This is a fork at Weekend followed by a chain, none of which are conditioned, so it is open.

5. **Friends over ⊥ Get Sleep | Traveling?** YES. The two paths are:

   - Friends over ← Weekend → Traveling → Get Sleep: a chain with Traveling conditioned, which blocks it.
   - Friends over → Eat exotic foods ← Traveling → Get Sleep: a collider at Eat exotic foods, unconditioned, so it is blocked.

6. **Modified network (remove Traveling→Get Sleep):**



7. **In modified network, Friends over ⊥ Get Sleep?** YES. All paths between Friends over and Get Sleep are blocked:

   - Any path via Weekend and Traveling ends at Traveling, which no longer connects to Get Sleep.
   - The only other candidate path is Friends over → Eat exotic foods ← Sick → Get Sleep, which is blocked by the unconditioned collider Eat exotic foods.

   Conditioning on Eat exotic foods (the collider) would open that path, making them dependent.

Name: Matt Krasnow
Collaborators and Resources: Chatgpt for formatting and latex upload