Assignment #5

Due: 7:59pm EST, April 2, 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 Categorical-Geometric Mixture Models, 25pts)

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

Specifically, we assume that there are a set of K parameters $p_k \in [0,1]$. To generate each observation n, we first choose which of those K parameters we will use based on the (unknown) overall mixing proportion over the components $\boldsymbol{\theta} \in [0,1]^K$, where $\sum_{k=1}^K \boldsymbol{\theta}_k = 1$. Let the (latent) $\mathbf{z}_n$ indicate which of the K components we use to generate observation n. Next we sample the observation $\mathbf{x}_n$ from a geometric distribution with parameter $p_{\mathbf{z}_n}$. This process can be written as:

$$\begin{aligned}\mathbf{z}_n &\sim \text{Categorical}(\boldsymbol{\theta})\\ \mathbf{x}_n &\sim \text{Geometric}(p_{\mathbf{z}_n})\end{aligned}$$

We encode observation n's latent component-assignment $\mathbf{z}_n \in \{0,1\}^K$ as a one-hot vector. Element indicator variables z_{nk} equal 1 if $\mathbf{z}_n$ was generated using component k.

A geometric distribution corresponds to the number of trials needed to get to the first success, if success occurs with probability p. Its PMF is given by $p(x_n|p_k) = (1-p_k)^{x_n-1}p_k$

1. **Intractability of the Data Likelihood** We are generally interested in finding a set of parameters p_k that maximize the data likelihood $\log p(\{\mathbf{x}_n\}_{n=1}^N | \{p_k\}_{k=1}^K)$. Expand the data likelihood to include the necessary sums over observations $\mathbf{x}_n$ and to marginalize out (via more sums) the latents $\mathbf{z}_n$. Why is optimizing this likelihood directly intractable?

2. **Complete-Data Log Likelihood** The complete data $D = \{(\mathbf{x}_n, \mathbf{z}_n)\}_{n=1}^N$ includes latents $\mathbf{z}_n$. Write out the complete-data negative log likelihood. Apply the "power trick"[a] and simplify your expression using indicator elements z_{nk}.

$$\mathcal{L}(\boldsymbol{\theta}, \{p_k\}_{k=1}^K) = -\ln p(D \,|\, \boldsymbol{\theta}, \{p_k\}_{k=1}^K).$$

   Note 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 topic variables $\mathbf{z}_n$ conditioned on the observed data $\mathbf{x}_n$ with fixed parameters, i.e $\mathbf{q}_n = p(\mathbf{z}_n|\mathbf{x}_n; \boldsymbol{\theta}, \{p_k\}_{k=1}^K)$.

   - **Part 3.A** 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.
   - **Part 3.B** Give an algorithm for calculating the expression for $\mathbf{q}_n$ found in Part 3.A for all n, given the observed data $\{\mathbf{x}_n\}_{n=1}^N$ and settings of the parameters $\boldsymbol{\theta}$ and $\{p_k\}_{k=1}^K$.

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 $\{p_k\}_{k=1}^K$.

   - **Part 4.A** Derive an expression for the expected complete-data log likelihood using $\mathbf{q}_n$.
   - **Part 4.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?
   - **Part 4.C** Find an expression for the $\{p_k\}_{k=1}^K$ that maximize the expected complete-data log likelihood. Why does this optimal $\{p_k\}_{k=1}^K$ make intuitive sense?

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

   (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 $p(\mathbf{z}_n; \boldsymbol{\theta}) = \prod_{k=1}^K \boldsymbol{\theta}_k^{z_{nk}}$.

Problem 1 (cont.)

6. Finally, implement your solution (see `T5_P1.py` for starter code). You are responsible for implementing the `expected_loglikelihood`, `e_step` and `m_step` functions. Test it out with data given 10 samples from 3 components with $p_1 = .1$, $p_2 = .5$, and $p_3 = .9$. How does it perform? What if you increase the number of samples to 1000 from each of the components? What if you change $p_2 = .2$? Hypothesize reasons for the differences in performance when you make these changes. You may need to record five to ten trials (random restarts) in order to observe meaningful insights.

Solution

1.

$$\log p(\{x_n\}_{n=1}^N|\{p_k\}_{k=1}^K)$$

$$= \sum_{n=1}^N \log p(x_n|\{p_k\}_{k=1}^K)$$

$$= \sum_{n=1}^N \log \sum_{k=1}^K p(x_n, z_n|\{p_k\}_{k=1}^K)$$

$$= \sum_{n=1}^N \log \sum_{k=1}^K p(z_n = k)p(x_n|z_n, \{p_k\}_{k=1}^K)$$

$$= \sum_{n=1}^N \log \sum_{k=1}^K \theta_k(1-p_k)^{x_n-1}p_k$$

Turns out, there are no analytic solutions here and the gradients are messy. That's why this likelihood directly intractable.

2.

$$\sum_{n=1}^N \log p(x_n, z_n)$$

$$= \sum_{n=1}^N \log p(x_n|z_n)p(z_n)$$

$$= \sum_{n=1}^N \sum_{k=1}^K z_{nk} \log(\theta_k(1-p_k)^{x_n-1}p_k)$$

3. (a)

$$q_{nk} = p(z_n = k|x_n) \propto \theta_k(1-p_k)^{x_n-1}p_k$$

If we standardize it, we have

$$q_{nk} = p(z_n = k|x_n) = \frac{\theta_k(1-p_k)^{x_n-1}p_k}{\sum_{k=1}^K \theta_k(1-p_k)^{x_n-1}p_k}$$

(b) First, we randomly initialize the parameters p_k and θ_k, and use them to calculate q_n. After the first step, we will have new parameters from the maximization step for calculate new q_n for the following steps.

4. (a)

$$\sum_n E_z(\log p(x_n, z_n))$$

$$= \sum_n E_z(\log p(x_n|z_n)p(z_n))$$

$$= \sum_n \sum_k (p(z_n = k|x_n) \log p(x_n|z_n)p(z_n))$$

$$= \sum_n \sum_k (p(z_n = k|x_n) \log((1-p_k)^{x_n-1}p_k\theta_k))$$

$$= \sum_n \sum_k (q_{nk} \log((1-p_k)^{x_n-1}p_k\theta_k))$$

(b) For θ, we can ignore other independent terms and consider

$$\sum_n \sum_k (q_{nk} \log \theta_k)$$

Using Lagrangian multiplier, we have

$$L(\theta) = \sum_n \sum_k (q_{nk} \log \theta_k) - \lambda \sum_K \theta_k$$

Taking partial derivatives to λ and θ_k's, we get K+1 equations.

$$\sum_k \theta_k = 1, k = 1, 2, ...K$$

$$\sum_n q_{nk} \frac{1}{\theta_k} = \lambda$$

And I get the result

$$\theta_k = \frac{\sum_n q_{nk}}{N}$$

This is intuitive because it tells you what every single data point believes about the likeliness of a given cluster k.

(c) Similarly, consider only

$$\sum_n \sum_k (q_{nk} \log(1 - p_k)^{x_n - 1} p_k)$$

Trying to find a maximum p_k for each k, I get

$$p_k = \frac{\sum_n q_{nk}}{\sum_n q_{nk} x_n}$$

This is intuitive because it the isreciprocal the weighted average (weighted by how probable a given x belongs to a cluster) of x_n and we know the expectation for a simple geometric distribution is

$$E(x) = \frac{1}{p}$$

, which is consistent to our result.

5. Now q_n becomes one hot vectors based on our knowledge of the cluster. The results in the above question now become

$$\theta_k = \frac{\sum_n z_{nk}}{N}$$

$$p_k = \frac{1}{\sum_n z_{nk} x_n}$$

Then we should consider the generative classification problem and seek for $k \in \{1, ..., K\}$ which maximize the posterior probability

$$p(x|z = k)p(z = k) = (1 - p_k)^{x-1} p_k \theta_k$$

6. Here are 5 randoms runs for $n = 10$, $p_1, p_2, p_3 = 0.1, 0.5, 0.9$.
Run1:
time elapsed: 0.0891561508178711 seconds

```
Final probabilities: [0.1368749 0.1368749 0.7397984]
Accuracy: 0.3333333333333333
Run2:
Complete, time elapsed: 0.0285186767578125 seconds
Final probabilities: [0.04208856 0.44636983 0.52317011]
Accuracy: 0.4
Run3:
Complete, time elapsed: 0.07289385795593262 seconds
Final probabilities: [0.05201845 0.39704115 0.94127628]
Accuracy: 0.5666666666666667
Run4:
Complete, time elapsed: 0.042897939682006836 seconds
Final probabilities: [0.1046441 0.46328238 0.9276385 ]
Accuracy: 0.6
Run5:
Complete, time elapsed: 0.00302886962890625 seconds
Final probabilities: [0.1729419 0.64299275 0.88870058]
Accuracy: 0.5333333333333333
```

Here are 5 randoms runs for $n = 1000$, $p_1, p_2, p_3 = 0.1, 0.5, 0.9$.

```
Run1:
Complete, time elapsed: 0.8310668468475342 seconds
Final probabilities: [0.09030399 0.4487585 0.87113286]
Accuracy: 0.6653333333333333
Run2:
Complete, time elapsed: 0.8235440254211426 seconds
Final probabilities: [0.11297502 0.64965728 0.89087495]
Accuracy: 0.652
Run3:
Complete, time elapsed: 6.020646810531616 seconds
Final probabilities: [0.0960554 0.47482722 0.91432466]
Accuracy: 0.6693333333333333
Run4:
Complete, time elapsed: 0.5890040397644043 seconds
Final probabilities: [0.12339972 0.69549015 0.93860294]
Accuracy: 0.6463333333333333
Run5:
Complete, time elapsed: 0.23727202415466309 seconds
Final probabilities: [0.0958631 0.45002302 0.98557814]
Accuracy: 0.654
```

Here are 5 randoms runs for $n = 1000$, $p_1, p_2, p_3 = 0.1, 0.2, 0.9$.

```
Run1:
Complete, time elapsed: 2.032442808151245 seconds
Final probabilities: [0.10075082 0.21010593 0.91730553]
Accuracy: 0.6566666666666666
Run2:
Complete, time elapsed: 1.177523136138916 seconds
Final probabilities: [0.10326203 0.13281749 0.77350411]
Accuracy: 0.6393333333333333
Run3:
Complete, time elapsed: 2.3584179878234863 seconds
```

Final probabilities: [0.10778701 0.24790742 0.92930897]
Accuracy: 0.6156666666666667
Run4:
Complete, time elapsed: 0.2372140884399414 seconds
Final probabilities: [0.08598604 0.20103908 0.88942227]
Accuracy: 0.65
Run5:
Complete, time elapsed: 0.23415088653564453 seconds Final probabilities: [0.05430096 0.12189632 0.77447492] Accuracy: 0.5473333333333333

Discussion: when only include 10 data samples, the elapsed time is definitely smaller, and the accuracy is relatively low and fluctuates a lot with different data (accuracy between $\tilde{0}.3$-$\tilde{0}.6$). Including 1000 data samples significantly increases the accuracy and also decrease the fluctuation of the result (around 0.65). It is intuitive when increasing data size, the accuracy increase, because more data gives more information to the data and make it perform better. If there is only 10 points, it is not surprising the model is over-fitted to the data. And the elapsed time definitely increases because there are more data.

Changing p_2 to 0.2 slightly decrease the accuracy(here I still take 1000 points). That is because now p_1 and p_2 become closer, and the model can more easily confuse these to categories.

Problem 2 (PCA, 15 pts)

For this problem you will implement PCA from scratch. Using `numpy` to call SVDs is fine, but don't use a third-party machine learning implementation like `scikit-learn`.

We return to the MNIST data set from T4. You have been given representations of 6000 MNIST images, each of which are 28×28 greyscale handwritten digits. Your job is to apply PCA on MNIST, and discuss what kind of structure is found.

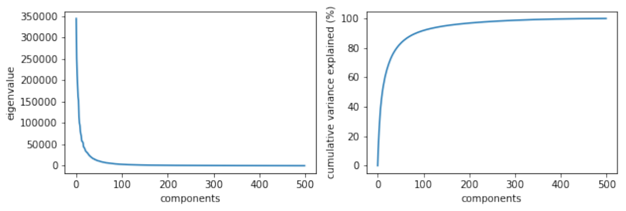
The given code in `T5_P2.py` loads the images into your environment. File `T5_P2_Autograder.py` contains a test case to check your cumulative proportion of variance.

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.
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 all 11 plots in your PDF submission.
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.

Include your plots in your PDF. There may be several plots for this problem, so feel free to take up multiple pages.

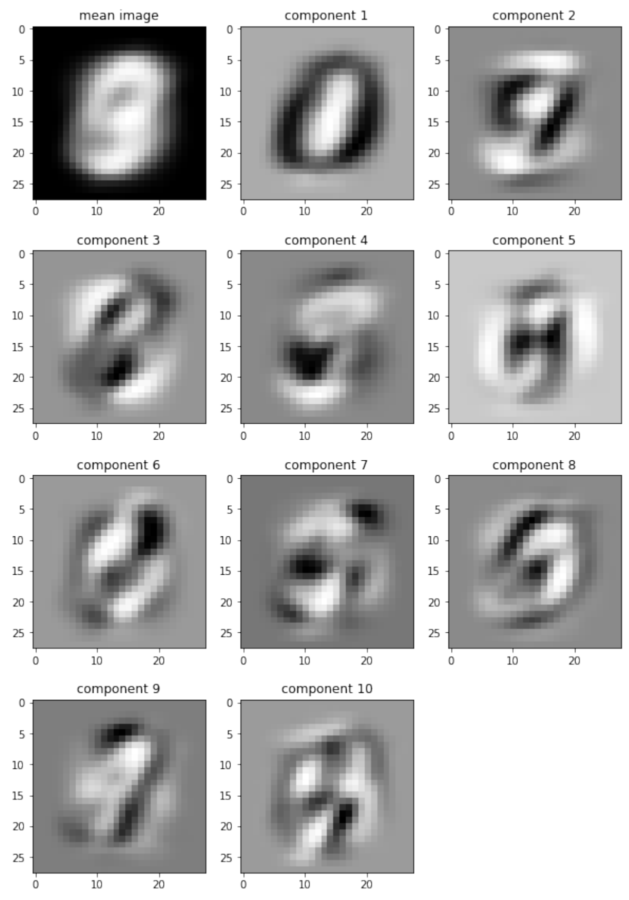
Solution

1. The figure is shown below. The first 500 components explained 99.94% of the variance. The cumulative variance increases fast when k is small, and as k grows larger, the variance increases slower. The first 100 components explains more than 90% of the variances.



2. The figures are shown below. For PCA result, we cannot easily see a identifiable '0-9' digit from the components, especially for components larger than 3. Every component look like a combination of a few different digits, while in K-means we can see more identifiable. Another interesting feature is we see very obvious negative areas(white areas) in our PCA analysis, which is not seen in Kmeans, that is because in PCA the final result is a combination of different components, but Kmeans is a simple classification.

   The similar thing to Kmeans is that the result is relatively 'grey', which is caused by subtracting the mean. And the patterns are more or less similar to the original '0-9' digits.



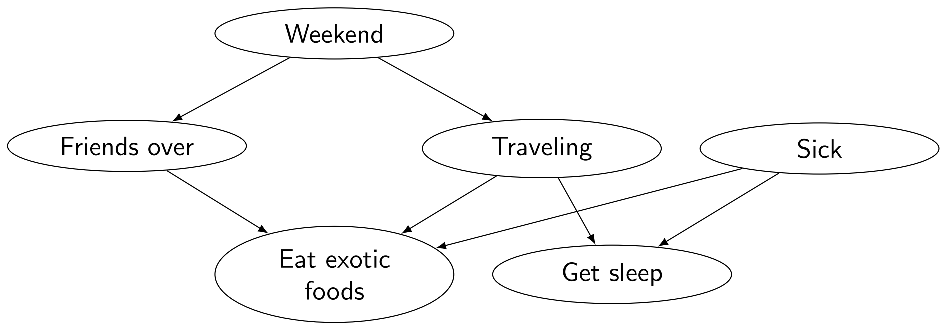
3. Reconstruction error on the dataset using the mean image: 3436023.412205305. This is actually just variance of dataset.

   Reconstruction error using first 10 principal components: 1731315.3279392195. This is definitely better than the simple mean image because it contains extra information and the first 10 components should catch a lot of the variance of the data. The difference between this error and the error from the mean image is the contribution of the first 10 components.

   Compare to Kmeans: the sizes of the dataset in HW4 and this question are not exactly the same, but generally speaking when divide the objective function of Kmeans by the size of the data (since we also do the same thing here in reconstruction error) , the results for 5, 10, 15 clusters on the large dataset are 2818896,2543671 2374508, respectively, and a larger than the reconstruction error here. That means our first 10 components of PCA catch the features bettern than Kmeans when using those number of clusters.

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.

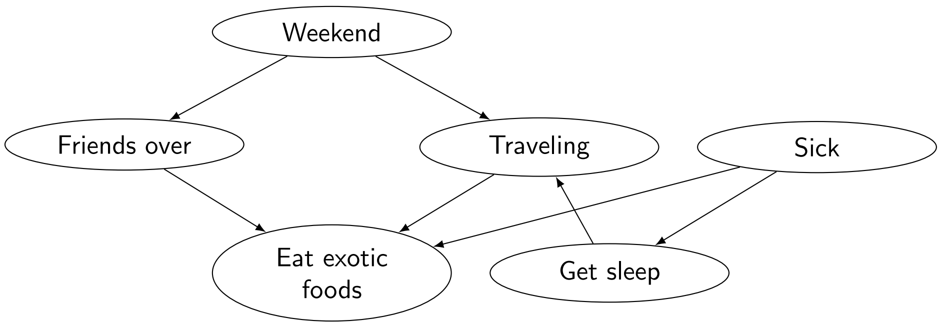
Example Answer: NO. The path from Friends over – Weekend – Traveling is not blocked following the d-separation rules. Thus, the two are not independent. Intuitively, this makes sense as if say we knew that the person was traveling, it would make it more likely to be the weekend. This would then make it more likely for the person to have friends over.

Actual Questions:

1. Is Sick $\perp$ Weekend? If NO, give intuition for why.
2. Is Sick $\perp$ Friends over | Eat exotic foods? If NO, give intuition for why.
3. Is Friends over $\perp$ Get Sleep? If NO, give intuition for why.
4. Is Friends over $\perp$ Get Sleep | Traveling? If NO, give intuition for why.
5. Suppose the person stops traveling in ways that affect their sleep patterns (as various famous people have done). 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).
6. 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. Yes.
2. No. If "eat exotic food" is observed, the the possibility of friends over will increase and of sick will decrease, in other words they will be anti-correlated.
3. No. If a person has friends over, it is more likely to be weekend, and then the person will likely to be traveling, then the person will more likely to sleep.
4. Yes.
5. Shown below.



Yes. If either Eat exotic foods is observed or travel is observed they will no longer be independent.

Name

Collaborators and Resources

Whom did you work with, and did you use any resources beyond cs181-textbook and your notes?

Calibration

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