6 Unsupervised Learning

Christopher Nota Last Monday at 11:57 AM

Comments are open.

S Add a comment...

Course: COMPSCI 589 Machine Learning, Spring, 2021

Instructor: Justin Domke

Assignment: 6

Group work policy: You are allowed to complete this homework in teams of at most 3 students. However, each student must submit their own individual .pdf and .zip files to Gradescope. List your team members at the beginning of your `report.pdf`. You may verbally discuss the assignment with course staff or students outside your group. Please also list any students or course staff (separately) at the beginning of your `report.pdf`. However, you may not *look, copy, or show* any part of another student's assignment. Copying any part of another assignment — even a single sentence or line of code — from anyone outside your team is considered plagiarism. We use sophisticated tools to detect this. Please do not do it.

Due date: May 4, 2021, 5:00 PM

Submission instructions:

- For this assignment, you should prepare your solutions in one of three formats:
  - Latex (any style)
  - Markdown
  - Jupyter notebook
- Regardless of how you prepared the solutions, you should export a single .pdf file that you upload to Gradescope. The .pdf should be submitted to Assignment 6: Unsupervised Learning. Most coding question will ask you to include your code as text in the solution .pdf.

- Additionally, you **must submit a .zip file** to Assignment 6: ZIP File in Gradescope. Your .zip file should contain four things:
  - `report.pdf` - Your report.
  - `report_src/` - A directory containing all source files for the report.
  - `code/` - A directory containing Python code for all parts of the assignment.
  - `code/run_me.py` - A single Python file that will generate all figures included in your report.
- If you use a Jupyter notebook, nothing changes. You still must put your code in external files, and you still must submit both a single .pdf and a .zip file containing the above components to the respective assignments in Gradescope.
- When you submit the .pdf to Gradescope, you must must mark page numbers for the different questions. We hate to do it, but we will penalize anyone who does not do this, as it creates a huge amount of difficulty for the graders.
- For the purpose of late days, the later of your two submissions will be considered the submission time for your assignment. E.g., if you submit your .pdf on time, but the .zip is two days late, the assignment will be considered two days late.

In this assignment, you will implement two simple versions of lossy image compression— one based on PCA, and the other on K-means. Each question is worth 10 points.

PCA

Suppose you are given a dataset of N samples $x^{(1)}, \cdots, x^{(N)}$, each of which has D dimensions. We could compress this dataset by seeking a set of N scalars $a_1, \cdots, a_N$ along with a single vector w with D dimensions. The approximation is that

$$x^{(n)} \approx a_n w.$$

Suppose we want the values a_n and w to minimize reconstruction error, i.e. we want

$$\min_w \min_a \frac{1}{N}\sum_{n=1}^{N} \|x^{(n)} - a_n w\|^2.$$

Let X be a matrix with $x^{(n)}$ on the nth row.

Question 1: Suppose that w is fixed. What is the value of a that minimizes reconstruction error

$$\arg\min_a \frac{1}{N}\sum_{n=1}^{N} \|x^{(n)} - a_n w\|^2?$$

Show your work.

Question 2: Suppose w is fixed. What is the reconstruction error if each a_n is set to the optimal value you found in the previous question, i.e.

$$\min_a \frac{1}{N}\sum_{n=1}^{N} \|x^{(n)} - a_n w\|^2?$$

Show your work.

Question 3: If you minimize the reconstruction error over both w and a, what is the optimal w? Give your answer in terms of the matrix X using eigenvalues and eigenvectors. Show your work.

4

Hint: You might as well assume that $\|w\| = 1$ since if this is not true, you could rescale a to make it true.

Hint: You might find it helpful to define the covariance matrix of the data, $C = \frac{1}{N}\sum_{n=1}^{N} x^{(n)} x^{(n)\top}$.

Question 4: Suppose that, in the training data, for all n,, $x_2^{(n)} = 2 \times x_1^{(n)}$, $x_4^{(n)} = 2 \times x_3^{(n)} \cdots$ and finally $x_D^{(n)} = 2 \times x_{D-1}^{(n)}$. (Assume D is even.)

Suppose you will project and reconstruct this data using a linear codebook, as described in lecture. What is the minimum number of dimensions you will need in your code such that the data can be reconstructed without error? Explain in at most 4 sentences.

Hint: Thinking about the cases where $D = 2$ or $D = 4$ might help.

Faces.zip 231.2KB

Question 5: You are given a set of 100 images of faces. Each is 50x50 and grayscale, and so can be treated as a vector in $\mathbb{R}^{2500}$. The covariance matrix of the data is

$$C = \frac{1}{N} \sum_{n=1}^{N} (x^{(n)} - \bar{x})(x^{(n)} - \bar{x})^\top,$$

where $\bar{x} = \frac{1}{N} \sum_{n=1}^{N} x^{(n)}$. Note that $C \in \mathbb{R}^{2500 \times 2500}$. For each value of $k \in \{3, 5, 10, 30, 50, 100\}$, find the k eigenvectors associated with the k largest eigenvalues of C. Call these vectors $w_1, \cdots, w_k$. Project the data on to them, and call the compressed representations $y^{(1)}, \cdots, y^{(N)}$. Note that $y^{(n)} \in \mathbb{R}^k$ and

$$y_j^{(n)} = w_j \cdot x^{(n)}.$$

You can approximately reconstruct $x^{(n)}$ as

$$\hat{x}^{(n)} = \sum_{j=1}^{k} y_j^{(n)} w_j.$$

Show in your report the original `face.png` (download the image above) as well as the reconstruction obtained for each value of k.

~~For this question, you may use a package that computes eigenvalues and eigenvectors (e.g. `numpy.linalg.eigh`) but you may **not** use any package that computes PCA.~~ (Apologies for the confusion — YOU MAY USE SKLEARN'S PCA.) You may also not use any external libraries for image processing except for `numpy`, `sklearn`, and `matplotlib`.

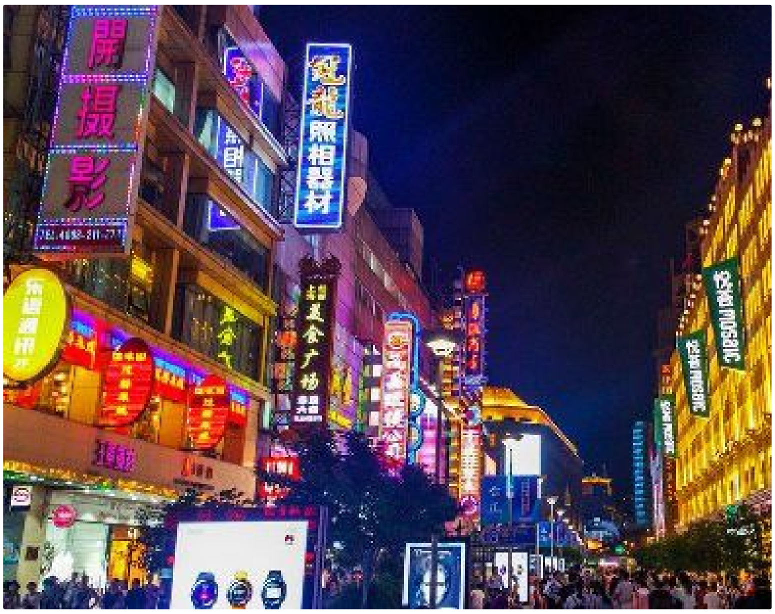
Hint: PNG images can be loaded using `matplotlib.mpimg`:

```
import matplotlib.image as mpimg img = mpimg.imread('face.png')
```

Question 6: Make a table showing the compression rate for each value of k. The compression rate is the amount of memory needed to store the compressed representation ($w_1, \cdots, w_k$ and $y^{(1)}, \cdots, y^{(N)}$) divided by the amount of memory used to store the original data. Represent all data, including the original image, using a 64-bit float.

K-Means

In the following questions, you will compress the following (single) image of width 393 and height 309:



For these questions, you *are permitted* to use sklearn for your implementation of k-means.

Question 7: K-means is an algorithm that splits the data into clusters. There are various different ways to choose the number of clusters. The "elbow" rule is a common heuristic. Explain it in at most 4 sentences.

Question 8: Another issue with K-means is that the final results depend on initialization. A possible solution to this problem is called K-means++. Briefly explain the idea behind this algorithm.

Question 9: You are given above an image of a shopping street. If you break this up into 3x3 chunks, there will be 13,493=131x103 total chunks. Since pixels are RBG, each can be interpreted as a vector in $\mathbb{R}^{27}$. Transform the dataset into a set of 13493 elements of length 27. Next, for each $k \in \{2, 5, 10, 25, 50, 100, 200, 1000\}$, apply K-means clustering, so that each 3x3 chunk is represented by a single integer. Reconstruct the original image from each compressed representation and show it. Show one reconstructed image for each value of k.

Question 10: For each value of k in the above question, compute the reconstruction error, the mean squared difference in intensity, averaged over all pixels, and color channels. Show your results as a table.

Question 11: How many total numbers are in your compressed representation for each value of k? Give your answer as a table.

2

Question 12: For each value of k compute the compression rate. Don't worry about bits, just assume that the original image uses 393x309x3 numbers and use your answer from the previous question. Give your answer as a table.