Homework 4: More Principal Component Analysis

About

Due

Monday 2/18/19, 11:59 PM CST

Goal

This homework focuses on familiarizing you with low-rank approximations and multi-dimensional scaling. In addition, you will work with the CIFAR-10 dataset, a popular benchmark dataset for most classification algorithms.

Additionally, it is intended to provide practice with finding and using publicly available libraries, an essential skill when applying machine learning techniques.

Code and External Libraries

The assignment can be done using any language.

You may use external libraries to perform PCA, as well as to compute euclidean distances.

For python check out:

- `PCA` from `sklearn.decomposition`
- `pdist` and `squareform` from `sklearn.spatial.distance`
- `euclidean_distances` from `sklearn.metrics`

For R, you can use `as.matrix(dist(m))` to generate a matrix of Euclidean distances between the rows of the matrix `m`.

You are expected to write your own code for Principal Coordinate Analysis.

Problems

Total points: 100

CIFAR-10 is a dataset of 32x32 images in 10 categories, collected by Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. It is often used to evaluate machine learning algorithms. You can download this dataset from https://www.cs.toronto.edu/~kriz/cifar.html (https://www.cs.toronto.edu/~kriz/cifar.html). You should combine the test and train sets (all the images) and separate them by category.

A. For each category, compute the mean image and the first 20 principal components. Plot the error resulting from representing the images of each category using the first 20 principal components against the category **as a bar graph** (refer to the **Procedures** section for clarifications).

B. Compute the distances between mean images for each pair of classes. Use principal coordinate analysis (refer to the **Procedures** section for clarifications) to make a 2D map of the means of each categories. (Follow procedure 7.2 on page 120 of the book)

C. Here is another measure of the similarity of two classes. For class A and class B, define E(A → B) to be the average error obtained by representing all the images of class A using the mean of class A and the first 20 principal components of class B (Refer to **Procedures** section for explicit definition). This should tell you something about the similarity of the classes. Now define the distance metric between classes to be (1/2)(E(A → B) + E(B → A)). Use principal coordinate analysis to make a 2D map of the classes. Compare this map to the map in the previous exercise – are they different? why?

Procedures

Outline

For all error calculations, it will be helpful to flatten your images. After compiling the data from the dataset, you will likely have it in a 4-D array with shape `(60000, 32,32,3)`. Flattening the images should give you a 2-D array with shape `(60000, 3072)`. This will make the following computations easier to understand.

Part A

1. For each class, find the mean image, and compute the first 20 principal components.
2. Now use the mean as well as the principle components to compute a low-dimensional reconstruction of each image in the class. **Hint:** Libary functions will come in handy here. Refer to section 7.1.2 and 7.1.3 for theory.
3. Now for each image, compute the squared difference between the original and reconstructed version, and sum this over all pixels over all channels. If you have flattened your images, this is simply the squared euclidean distance between the image vectors. Take the average of the value you computed above over all images in the class.
4. Plot the above value in the bar graph against its category/class label. You will submit this plot.

Part B

1. Compute a 10 x 10 distance matrix `D` such that `D[i,j]` is the **Euclidean distance** between the mean images of class `i` and class `j`. **Square the elements of this matrix** and write it out to a CSV file named *partb_distances.csv*. You will submit this file.
   Note: The order of the class labels is very important here, as this file will be autograded. Refer to this (./hw4_label_ordering.json) for the index-label mapping, and ensure yours matches.
2. Now you must perform multi-dimensional scaling with the squared distance matrix you have. Refer to the MDS section for details on how to do that.

3. Once you have computed the scaled points in 2-D space, plot the first component along the x-axis and component 2 along the y-axis of a scatter plot. You will submit this plot.

Part C

1. Just like in Part B, you will first compute a 10 x 10 distance matrix. However, here, `D[i,j]` will contain E(i → j). Let's define E(A → B).
    - E(A → B) = (E(A| B) + E(B|A))/2
    - To compute E(A|B), use the mean image of *class A* and the first 20 principal components of *class B* to reconstruct the images of *class A*
    - Once you have the reconstructed images, use the procedure described in steps 3 and 4 of Part A to compute the mean of the sum of pixel-wise squared difference between the reconstructed and original images.
    - Similarly compute E(B|A).
    - **Note: E(A|A) != 0**, as a sanity check.
2. Once you have computed `D`, write it out to a CSV file named *partc_distances.csv*. You will submit this file. Again, make sure the index-label ordering is correct in your matrix. **Note:** There is no need to square the values in `D` as they are already averaged square distances.
3. Perform MDS with this distance matrix, and once you have the scaled points in 2-D, plot the first component along the x-axis and component 2 along the y-axis of a scatter plot. You will submit this plot.

Principal Coordinate Analysis (MDS)

This procedure can be found on page 120 of the textbook. There are some minor typos in the textbook version. Refer to this procedure instead. For the following procedure, the set of points whose mutual distances you will start out with are the mean images of each class. **Note:** Be careful not to accidentally square your already squared distances matrices when implementing the second bullet point below.

Assume we have a matrix $\mathbf{D^{(2)}}$ consisting of squared differences between each pair of N points, and we wish to compute a set of points in *s* dimensions, such that the distances between these points are as similar as possible to the distances in $\mathbf{D^{(2)}}$.

- First form the centering matrix **A** as described in section 7.1.2 on page 118. $\mathbf{A} = \mathbf{I} - \frac{1}{N}\mathbf{1}\mathbf{1}^T$
- Now form $\mathbf{W} = -\frac{1}{2}\mathbf{A}\mathbf{D^{(2)}}\mathbf{A}^T$
- Next, form **U** and **Λ** such that **WU = UΛ**. These are respectively the eigenvectors and eigenvalues of **W**. Ensure that the entries of **Λ** are sorted in decreasing order. Notice that you need only the top *s* eigenvalues and their eigenvectors, and many packages can extract these quickly, instead of constructing all of them.
- Choose *s*, the number of dimensions you wish to represent. Form $\mathbf{\Lambda_s}$, the top left *s x s* block of **Λ**.
- Form $\mathbf{\Lambda_s}^{1/2}$, whose entries are the positive square roots of $\mathbf{\Lambda_s}$. Construct $\mathbf{U_s}$, the matrix consisting of the first *s* columns of **U**.
- Finally, compute $\mathbf{Y} = \mathbf{U_s}\mathbf{\Lambda_s}^{1/2} = [\mathbf{v_1}, \ldots, \mathbf{v_N}]$ **. This is the set of points you must plot.**

Submission

Submission will be through gradescope (https://www.gradescope.com):

Your submission for this homework should include:

1. PDF report with the following to the **HW4 Report** portal:
    1. Page 1: **(10 points)** A plot of the mean image for each class.
    2. Page 2: **(15 points)** A plot of the average sum of squared pixel-wise difference between the reconstructed and original images of each class **as a bar graph** vs. the class label.
    3. Page 3: **(25 points)** The 2D scatter plot obtained after performing principal coordinate analysis using euclidean distance.
    4. Page 4: **(25 points)** The 2D scatter plot obtained after performing principal coordinate analysis using the similarity metric in part C.
2. Code submission via gradescope to the **HW4 Code** portal:
    1. **(5 points)** Submit your code as a zipped file.
    2. **(10 points)** Submit a CSV containing the distance matrix between the mean images of each class as a CSV file named partb_distances.csv
    3. **(10 points)** Submit a CSV containing the distance matrix between the mean images of each class as a CSV file named partc_distances.csv