Assginment 4
Dimension Reduction

May 16, 2018

Due: May 28, 2018 11:59 PM PDT
Late policy: 5% of the total received points will be deducted on the first day past due. Every 10% of the total received points will be deducted for every extra day thereafter.

Introduction

This assignment includes two kinds of principle component analyses (PCA) for dimension reduction. We start from the relation between PCA and singular value decomposition (SVD). Then, we move on to the limitation of PCA and its solution (robust PCA).

More specifically, students are required to read 3 on-line articles and work on a short proof question. Task 2 and 3 are almost the same except for a modification of the YaleB dataset to occlude 500 images.

Datasets

Throughout this assignment, we use the extended and cropped version of the Yale face recognition dataset. All subjects' faces are cropped into the same size of images. Table 1 describes the dataset.

	# features	Data source
Extended Yale B (YaleB)	168x192 16-bit gray-scale	The extended Yale face recognition dataset B cropped version `http://pages.ucsd.edu/~ztu/courses/CroppedYale.zip`
Description	The cropped YaleB dataset includes 38 human subjects in 38 folders. Each folder has 64 pictures of the same person. In this assignment, we use only the first 20 subjects, which provides 1280 images. After a random shuffling, we select the first 1024 images as the training set and the rest 256 images as the test set.	

Table 1: INTC dataset description.

1 (30 points) Task 1: PCA and SVD

Reading for PCA

Read through the PCA article on Wikipedia before the “Further considerations” session:
`https://en.wikipedia.org/wiki/Principal_component_analysis`

Read through the best answer to the question trying to explain PCA to someone’s grandmother on StackExchange:
`https://ez2o.co/7vs8W`
Denis provided an vivid explanation with impressive animations which is very worth-reading.

Problem formulation

The problem formulation of PCA is to find a projection matrix for an input matrix so that the projected low-rank matrix can have minimal recovery error. Equation 1 shows the original problem formulation of PCA.

$$\begin{aligned} min_{P,\tilde{X}} &= ||X - \tilde{X}P^T||_2^2 \\ s.t \quad P^T P &= I, \ \texttt{and} \ rank(\tilde{X}) \leq k, \end{aligned} \tag{1}$$

where X is the input matrix, which does not has to be a square matrix, $\tilde{X}$ is a low-rank approximation of X, P is the projection matrix, I is an identity matrix, and k is the desired rank of the projected matrix.

Equation 1 can be proven to be mathematically equivalent with equation 2 (We skip this proof in the assignment.) This optimization is also called diagonalization in linear algebra since the resulting matrix $D = P^T X^T X P$ is a diagonal matrix.

$$\begin{aligned} & max_P P^T C P \\ s.t \quad & p_i^T p_j = 1_{(i=j)}, \end{aligned} \tag{2}$$

where p_i and p_j are the column vectors in the projection matrix P, and C is the covariance matrix of X as shown in equation 3. Since a covariance matrix can be diagonalize because it is a symmetric square matrix. (Note that we simplified the definition of a covariance matrix. In fact, you should remove the mean of each example first, and then divide the entire covariance matrix with the number of examples. We ignore these 2 step for the simplicity of the assignment.)

$$C = X^T X \tag{3}$$

Proof of the relation between PCA and SVD

Given equation 4 as the singular value decomposition of input feature matrix X, please prove $P = V$ and $D = S^2$.

$$X = USV^T, \tag{4}$$

where S is the diagonal eigenvalue matrix, and U and V are the left and right eigenvector matrix, respectively.
Hint: $S^T S = S^2$ because S is a diagonal matrix.
Hint: Every column vector in U(or V) is orthogonal to other column vectors in U(or V). That means $U^T U = I$ and $V^T V = I$.

Here is the step-by-step instruction to prove the relation.

1. Plug the result of SVD in equation 4 into equation 3.

2. Make use of the orthogonality of the left singular matrix U to simplify the right hand side (RHS). Your left hand side should remain as a C.

3. Make use of the power index property of the diagonal matrix S to simplify the RHS.

4. Make use of the orthogonality of the right singular matrix V to move V and V^T to LHS of the equation. You will get only S^2 on the RHS.

5. Inspect the result you have got and compare it with the target equation $D = P^T X^T X P$. You should find out the proof is done.

In your report, Please write down the result of every step neatly.

Reading for Robust PCA

We have already seen the limitation of PCA, and the solution, which is the RPCA, on class. Here, we provide a online article for further reading about RPCA.

Read through short article "ROBUST PRINCIPAL COMPONENT ANALYSIS VIA ADMM IN PYTHON". Although we choose to use inexact ALM in this assignment, the author pinpointed the limitation of PCA with two figures and provided a list of python implementations of RPCA.
`https://ez2o.co/8vzYC`

2 (35 points) Task 2: Face Recognition

In this task, we perform three subtasks for face recognition on YaleB dataset.

Data Preprocessing

We need to downsize the dataset and remove unnecessary files from the YaleB dataset.

1. Manually downsize the dataset by remove subfolders from yaleB22 to yaleB39. Doing this gives us 20 sub-folders of the first 20 subjects.
2. Make sure you are in the CroppedYale folder in the command line window.
3. Remove the log files by command line
   $ find . -name "*.LOG" -type f -delete
4. Remove the info files by command line
   $ find . -name "*.info" -type f -delete
5. Remove the DEADJOE file by command line
   $ find . -name "DEADJOE" -type f -delete
6. Remove the ambient light images by command line
   $ find . -name "*Ambient*" -type f -delete

Now we have 1,280 images in total distributed in 20 classes. (Each subject is a class.) Use "SnippetForYaleB.ipynb" to read in the preprocessed dataset. It basically uses **sklearn.datasets.load_files(.)** to iterate over the subfolders of CroppedYale and streams PGM images into strings. Then, we use **map(.)** to map the **read_pgm2(.)** to the strings and get numpy ndarrays. All labels are automatically encoded into integers.

To remove the means in every example, we use **preprocessing.scale(X.astype(float),axis=1)**, where X is the dataset. We shuffle the 1,280 images and split the them into a training set of the first 1,024 images and a test set of the rest 256 images.

Face Recognition

We move on to use linear SVM of sklearn to classify the subset of YaleB. Each subtask costs 10 points.

1. **SVM scheme** Use sklearn library to train a linear SVM **svm.LinearSVC()** on the training set. The training should take roughly 5 minutes. After the training, use the **score(.)** function of linear SVM to calculate and report the test accuracy. (The expected test accuracy should be higher than 99%.)
2. **PCA-SVM scheme** Use the **PCA(n_components=300)** in sklearn library to reduce the dimension of the dataset X from (1280, 32256) to (1280, 300). Also, split the PCA-reduced dataset into 1024 training examples and 256 test examples. Calculate and report the test accuracy for

a second linear SVM. (It should take 10 seconds to train the PCA, and another 10 seconds to train the linear SVM. The expected test accuracy is around 97%.)

3. **RPCA-PCA-SVM scheme** Use the "SnippetForIALM.ipynb" to perform robust PCA on the dataset X to obtain the low rank matrix A and sparse matrix E. Then we use another **PCA(n_components=300)** to reduce the size of low rank matrix and get the RPCA-reduced dataset. Train and test a third linear SVM on the RPCA-reduced dataset. Report your test accuracy in your report. (It takes 7 minutes to train the robust PCA, 13 seconds to train another PCA, and 9 seconds to train the third linear SVM. The expected test accuracy should be higher than 99%.)

(5 points) In your report, briefly describe the pros and cons of PCA and robust PCA.

3 (35 points) Task 3: Occluded Face Recognition

In this task, we repeat the same face recognition process on an occluded YaleB dataset.

Assuming the 20-subject subset is still in your hard disk, use "SnippetForOcclude.ipynb" to randomly pick 500 images and erase the area of subject's eyes. (Note that the occluded images are not stored back to the hard disk, they exist only in memory as Numpy variables.)

Taking the partially occluded dataset, perform the same three sub-tasks in Task 2 again. Report the 3 test accuracies in your report (10 points for each subtask).

(5 points) In your report, briefly describe the pros and cons of PCA and robust PCA on the partially occluded dataset.

(**Bonus 10 points**) Without changing the **n_components** parameter of all PCA, can you improve the accuracy of the third subtask, which is a combination of RPCA-PCA-SVM? How about combining the low rank matrix A and the sparse error matrix E and fit the PCA on the combined data?

Requirement

Write a brief report summarizing your work and comparing the results. Zip your report together with necessary codes and submit it via TED.