

Stony Brook University
CSE512 – Machine Learning – Fall 19
Homework 6, Due: 10 Dec at 11:59PM

This homework contains 3 questions. The last two questions require programming. The maximum number of points is 100.

1 PCA via Successive Deflation [30 points]

Suppose we have a set of n data points $\mathbf{x}_1, \dots, \mathbf{x}_n$, where each \mathbf{x}_i is represented as a d -dimensional column vector. Assume that the data has been centeralized, i.e., having zero mean: $\frac{1}{n} \sum_{i=1}^n \mathbf{x}_i = \mathbf{0}$. Let $\mathbf{X} = [\mathbf{x}_1; \dots; \mathbf{x}_n]$ be the $(d \times n)$ matrix where column i is equal to \mathbf{x}_i . Define $\mathbf{C} = \frac{1}{n} \mathbf{X} \mathbf{X}^T$ to be the covariance matrix of \mathbf{X} , where $c_{ij} = \frac{1}{n} \sum_{l=1}^n x_{il} x_{jl} = \text{covar}(i, j)$.

Next, order the eigenvectors of \mathbf{C} by their eigenvalues (largest first), and let $\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_k$ be the first k eigenvectors. These satisfy

$$\mathbf{v}_i^T \mathbf{v}_j = \begin{cases} 0 & \text{if } i \neq j \\ 1 & \text{if } i = j \end{cases}$$

\mathbf{v}_1 is the first principal eigenvector of \mathbf{C} (the eigenvector with the largest eigenvalue), and as such satisfies $\mathbf{C} \mathbf{v}_1 = \lambda_1 \mathbf{v}_1$. Now define $\tilde{\mathbf{x}}_i$ as the orthogonal projection of \mathbf{x}_i onto the space orthogonal to \mathbf{v}_1 :

$$\tilde{\mathbf{x}}_i = (\mathbf{I} - \mathbf{v}_1 \mathbf{v}_1^T) \mathbf{x}_i$$

Finally, define $\tilde{\mathbf{X}} = [\tilde{\mathbf{x}}_1; \dots; \tilde{\mathbf{x}}_n]$ as the **deflated matrix** of rank $d - 1$, which is obtained by removing from the d -dimensional data the component that lies in the direction of the first principal eigenvector:

$$\tilde{\mathbf{X}} = (\mathbf{I} - \mathbf{v}_1 \mathbf{v}_1^T) \mathbf{X}$$

1. [7 points] Show that the covariance of the deflated matrix,

$$\tilde{\mathbf{C}} = \frac{1}{n} \tilde{\mathbf{X}} \tilde{\mathbf{X}}^T$$

is given by

$$\tilde{\mathbf{C}} = \frac{1}{n} \mathbf{X} \mathbf{X}^T - \lambda_1 \mathbf{v}_1 \mathbf{v}_1^T$$

(Hint: Some useful facts: $(\mathbf{I} - \mathbf{v}_1 \mathbf{v}_1^T)$ is symmetric, $\mathbf{X} \mathbf{X}^T \mathbf{v}_1 = n \lambda_1 \mathbf{v}_1$, and $\mathbf{v}_1^T \mathbf{v}_1 = 1$. Also, for any matrices \mathbf{A} and \mathbf{B} , $(\mathbf{A} \mathbf{B})^T = \mathbf{B}^T \mathbf{A}^T$.)

2. [7 points] Show that for $j \neq 1$, if \mathbf{v}_j is a principal eigenvector of \mathbf{C} with corresponding eigenvalue λ_j (that is, $\mathbf{C} \mathbf{v}_j = \lambda_j \mathbf{v}_j$), then \mathbf{v}_j is also a principal eigenvector of $\tilde{\mathbf{C}}$ with the same eigenvalue λ_j .
3. [8 points] Let \mathbf{u} be the first principal eigenvector of $\tilde{\mathbf{C}}$. Explain why $\mathbf{u} = \mathbf{v}_2$. (You may assume \mathbf{u} is unit norm.)
4. [8 points] Suppose we have a simple method f for finding the leading eigenvector and eigenvalue of a positive-definite matrix, denoted by $[\lambda, \mathbf{u}] = f(\mathbf{C})$. Write some pseudocode for finding the first k principal basis vectors of \mathbf{X} that only uses the special f function and simple vector arithmetic.

(Hint: This should be a simple iterative routine that takes only a few lines to write. The input is \mathbf{C}, k , and the function f , the output should be \mathbf{v}_j and λ_j for $j \in 1, \dots, k$)

2 Action recognition with CNN (35 points + 10 bonus)

In this question, you will train a convolutional neural network (CNN) to classify images and videos using Pytorch. We use the UCF101 data (see <http://crcv.ucf.edu/data/UCF101.php>). There are also 10 classes of data in this homework but the data and the number of classes are different from those of Homework 4. Each clip has 3 frames and each frame is $64 * 64$ pixels. The labels of train and validation clips are provided in *hw6_data.mat* inside the directory Question2.

You will first train a CNN for action classification for each image. You will then improve the network architecture and submit the classification results on the test data to Kaggle. Then, you will train a CNN using 3D convolution for a set of video frames (rather than for individual frames), and submit your results to Kaggle.

The detail instructions and questions are in the jupyter notebook *Action_CNN.ipynb*. In this file, there are 8 ‘Todos’ spots for you to fill. The score of each Todo is specified at the spot. For the 5th and 8th Todos, you need to submit CSV result files to Kaggle. The results would be evaluated by Categorization Accuracy. For the 5th Todo, submit to <https://www.kaggle.com/c/cse512hw6image>. For the 8th Todo, submit to <https://www.kaggle.com/c/cse512hw6video>.

We will maintain a leader board for each Kaggle competition, and the top three entries at the end of the competition (official assignment due date) will receive 10 bonus points. Any submission that rises to top three after the assignment deadline is not eligible for bonus points. The ranking will be based on the Categorization Accuracy. Marks for these questions will be scaled according to the ranking on the **Private** Leaderboard. To prevent exploiting test data, you are allowed to make a maximum of 2 submissions per 24 hours. Your submission will be evaluated immediately and the leader board will be updated.

Environment setting

Please make a *./data* folder under the same directory with the *Action_CNN.ipynb* file. Put data *trainClips*, *valClips*, *testClips* and *hw6_data.mat* from the *Question2* directory under *./data*.

We recommend using virtual environment for the project. If you choose not to use a virtual environment, it is up to you to make sure that all dependencies for the code are installed globally on your machine. To set up a virtual environment, run the following in the command-line interface:

```
cd your_hw6_folder
sudo pip install virtualenv          # This may already be installed
virtualenv .env                      # Create a virtual environment
source .env/bin/activate              # Activate the virtual environment
pip install -r requirements.txt       # Install dependencies
# Note that this does NOT install TensorFlow or PyTorch,
# which you need to do yourself.

# Work (hard) on the assignment
# ... and when you're done:
deactivate                            # Exit the virtual environment
```

Note that every time you want to work on the assignment, you should run ‘source .env/bin/activate’ (from within your hw6 folder) to re-activate the virtual environment, and deactivate again whenever you are done.

3 Action Classification Using RNN (35 points)

In this section, you will train recurrent neural networks (RNNs) to classify human actions. RNNs are designed handle sequential data.

For human action recognition, you will be using skeleton data that encodes the 3D locations of 25 body joints. The data is collected by Kinect v2. There are 10 different action classes. There are 4000 training

sequences, 800 validation sequences, and 1000 test sequences. Each sequence has 15 frames, each frame is a 75-dimension vector (the xyz positions of 25 joints). Data and the Jupyter notebook for this question are provided inside the directory Question3.

You will first train an LSTM for action classification. Then try to improve the network architecture and attach your results with the jupyter notebook. Also add the hyper-parameters explored.

The detailed instructions and questions are in the jupyter notebook *RNN_ActionClassify.ipynb*. In this file, there are 4 ToDo locations for you to fill. The score of each ToDo is specified at the spot.

You will need to install the following extra packages:

```
pip install h5py
pip install git+https://github.com/pytorch/tnt.git@master
```

4 What to submit?

4.1 Blackboard submission

You will need to submit both your code and your answers to questions on Blackboard. Put the answer file and your code in a folder named: SBUID.FirstName.LastName (e.g., 10947XXXX.lionel.messi). Zip this folder and submit the zip file on Blackboard. Your submission must be a zip file, i.e, SBUID.FirstName.LastName.zip. The answer file should be named: answers.pdf. The first page of the answers.pdf should be the filled cover page at the end of this homework. The remaining of the answer file should contain:

1. Answers to Question 1.
2. Completed Ipython notebooks *ActionCNN.ipynb* and *RNN_ActionClassify.ipynb*
3. For Question 2, also submit Prediction csv files which you submitted to Kaggle.
4. For Question 3, mention the best validation performance in the report, and submit the prediction files on the test and validation set. For the validation set, submit a single prediction file which results in the best performance.

4.2 Report

You can use Latex or MS Word for preparing the report; submit report in pdf format. Report should be typed. Don't submit scans/pics of handwritten answers.

4.3 Kaggle submission

No Kaggle submission required for this homework.

5 Cheating warnings

Don't cheat. You must do the homework yourself, otherwise you won't learn. You cannot ask and discuss with students from previous years. You cannot look up the solution online.

Cover page for answers.pdf
CSE512 Fall 2019 - Machine Learning - Homework 6

Your Name:

Solar ID:

NetID email address:

Names of people whom you discussed the homework with: