Stony Brook University
CSE512 – Machine Learning – Spring 18
Homework 4, Due: Wed April 25, 2018, 11:59pm

This homework contains 3 questions. The last question requires programming. The maximum number of points is 100.

1 Manual calculation of one round of EM for a GMM [30 points]

(Extended version of: Murphy Exercise 11.7) In this question we consider clustering 1D data with a mixture of 2 Gaussians using the EM algorithm. You are given the 1-D data points $x = [1 \;\; 10 \;\; 20]$.

M step

Suppose the output of the E step is the following matrix:

$$R = \begin{pmatrix} 1 & 0 \\ 0.3 & 0.7 \\ 0 & 1 \end{pmatrix}$$

where entry $R_{i,c}$ is the probability of observation x_i belonging to cluster c (the responsibility of cluster c for data point i). You just have to compute the M step. You may state the equations for maximum likelihood estimates of these quantities (which you should know) without proof; you just have to apply the equations to this data set. You may leave your answer in fractional form. Show your work.

1. *[3 points]* Write down the likelihood function you are trying to optimize.
2. *[6 points]* After performing the M step for the mixing weights π_1, π_2, what are the new values?
3. *[6 points]* After performing the M step for the means μ_1 and μ_2, what are the new values?
4. *[6 points]* After performing the M step for the standard deviations σ_1 and σ_2, what are the new values?

E step

Now suppose the output of the M step is the answer to the previous section. You will compute the subsequent E step.

1. *[3 points]* Write down the formula for the probability of observation x_i belonging to cluster c.
2. *[6 points]* After performing the E step, what is the new value of R?

2 PCA via Successive Deflation [30 points]

(Adapted from Murphy Exercise 12.7)

Suppose we have a set of n data points $\mathbf{x}_1, \ldots, \mathbf{x}_n$, where each x_i is a d-dimensional column vector. Let $\mathbf{X} = [\mathbf{x}_1; \ldots; \mathbf{x}_n]$ be the $(d \times n)$ matrix where the i^{th} column is $\mathbf{x}_i$. Define $\mathbf{C} = \frac{1}{n}\mathbf{X}\mathbf{X}^T$ to be the covariance matrix of $\mathbf{X}$, where $c_{ij} = \sum_{l=1}^{n} x_{il}x_{jl} = covar(i, j)$.

Next, order the eigenvectors of $\mathbf{C}$ by their eigenvalues (largest first), and let $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_k$ 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 $\tilde{\mathbf{X}} = [\tilde{\mathbf{x}}_1; \ldots; \tilde{\mathbf{x}}_n]$ 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. [5 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* $\mathbf{B}$*,* $(\mathbf{AB})^T = \mathbf{B}^T \mathbf{A}^T$*.)*

2. [5 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. [5 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. [5 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* $j \in 1, \cdots, k$*)*

3 Programming Question (Generative Adversarial Networks) [40 points]

In this section, you will train generative adversarial networks (GAN) to generate images using PyTorch. We use the MNIST data which is 60,000 training and 10,000 test images. Refer to the *jupyter notebook* for details.

You will first train a GAN for generating new images. Then try to improve the network architecture and attach your results with the jupyter notebook. Also add the hyper-parameters explored.

The detail instructions and questions are in the jupyter notebook $GAN.ipynb$. In this file, there are 7 "To-Do" locations for you to fill. The score of each To-Do is specified at the spot.

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_hw4_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 on the assignment for a while ...
# ... 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 hw4 folder) to re-activate the virtual environment, and deactivate again whenever you are done.

4 What to submit?

4.1 Blackboard submission

For question 1 & 2, please put everything in one single pdf file and submit it on Blackboard, please include your name and student ID in the first page of the pdf file. For question 3, submit the jupyter notebook files $GAN.ipynb$ with your answers filled at the 'To Do' locations. Put the PDF file and your jupyter notebook file in a folder named: SUBID_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, SUBID_FirstName_LastName.zip.

5 Cheating warnings

Don't cheat. You must do the homework yourself, otherwise you won't learn. You must use your SBU ID as your file name for the competition.