CS446: Machine Learning **Spring 2018**

Machine Problem 9

Handed Out: Jan. 16, 2018 *Due: Apr. 5, 2018 (11:59 AM Central Time)*

Note: The assignment will be autograded. It is important that you do not use additional libraries, or change the provided functions input and output.

Part 1: Setup

- Remote connect to an EWS machine.

```
ssh (netid)@remlnx.ews.illinois.edu
```

- Load python module, this will also load pip and virtualenv

```
module load python/3.4.3
```

- Reuse the virtual environment from mp1.

```
source ~/cs446sp_2018/bin/activate
```

- Copy mp9 into your svn directory, and change directory to mp9.

```
cd ~/(netid)
svn cp https://subversion.ews.illinois.edu/svn/sp18-cs446/_shared/mp9 .
cd mp9
```

- Install the requirements through pip.

```
pip install -r requirements.txt
```

- Create data directory and Download the data into the data directory.

```
mkdir data
wget --user (netid) --ask-password \
https://courses.engr.illinois.edu/cs446/sp2018\
secure/assignment9_data.zip  -O data/assignment9_data.zip
```

- Unzip assignment9_data.zip

```
unzip assignment9_data.zip -d data/
```

- Prevent svn from checking in the data directory.

```
svn propset svn:ignore data .
```

Part 2: Exercise

In this exercise we will implement and fit a Gaussian mixture model to the MNIST dataset to predict the digit label given an image. In `main.py`, an example pipeline is provided for you.

Part 2.1 Implementation

- **Reading in data.** In `utils/io_tools.py`, we will fill in one function for reading in the dataset. The datasets consist of vectorized images and the corresponding label.

  The format is comma separated, and the first column is the label: (label, pixel-11, pixel-12, ...).

  There are three csv files: the first, `simple_test.csv`, contains data generated from 2 Gaussians with different means, with labels 0 and 1 (this is useful for debugging purposes). The other two, `mnist_train.csv` and `mnist_test.csv`, are the train/test sets for the MNIST digits dataset.

- **Mixture model implementation.** In `models/gaussian_mixture_model.py`, we will implement the Gaussian mixture model. The model will support the following operations:

  - **Fit** This will run the EM algorithm on an unlabelled dataset to estimate the Gaussian components (mean, covariance, mixing ratio).
  - **Supervised Fit** This will run the EM algorithm on a labelled dataset to estimate the Gaussian components. Once you fit the model on a dataset, you will associate a class label to each of the Gaussians using majority voting (i.e. if most the vectors associated with Gaussian 1 are the digit '3', then that Gaussian will be labelled as '3').
  - **Supervised Predict** You will then use this model to make class predictions on a testing dataset.

We will grade `models/gaussian_mixture_model.py` and `utils/io_tools.py`.

Part 3: Writing Tests

In `test.py` we have provided basic test-cases. Feel free to write more. To test the code, run

```
nose2
```

Part 4: Submit

Submitting the code is equivalent to committing the code. This can be done with the follow command:

```
svn commit -m "Some meaningful comment here."
```

Lastly, double check on your browser that you can see your code at

```
https://subversion.ews.illinois.edu/svn/sp18-cs446/(netid)/mp9/
```