Exercise: Anomaly Detection and Recommender Systems

Overview:

In this exercise, you will implement the anomaly detection algorithm and apply it to detect failing servers on a network. In the second part, you will use collaborative filtering to build a recommender system for movies.

List of Files for this assignment:

ex8.py – main entry script for first part of exercise
ex8_cofi.py - main entry script for second part of exercise
ex8data1.mat - First example Dataset for anomaly detection
ex8data2.mat - Second example Dataset for anomaly detection
ex8_movies.mat - Movie Review Dataset
ex8_movieParams.mat - Parameters provided for debugging
movie_ids.txt - List of movies
ex8utils.py - contains the following functions:

- `multivariateGaussian` - Computes the probability density function for a Gaussian distribution
- `checkCostFunction` - Gradient checking for collaborative filtering
- `computeNumericalGradient` - Numerically compute gradients

ex8modules.py - the script that needs to be filled in by you for this assignment. It contains the following functions related to anomaly detection:

- `estimateGaussian` - Estimate the parameters of a Gaussian distribution with a diagonal covariance matrix
- `selectThreshold` - Find a threshold for anomaly detection

ex8modules_cofi.py - the script that needs to be filled in by you for this assignment. It contains the following function related to collaborative filtering:

- `cofiCostFunc` - The cost function for collaborative filtering.

What you should do:

ex8modules.py and ex8modules_cofi.py contains functions that are not yet implemented. Your task is to implement those functions by filling in "YOUR CODE HERE" sections. The details can be explained as follows:

[Section 1: Anomaly detection]

In this exercise, you will implement an anomaly detection algorithm to detect anomalous behavior in server computers. Our example case consists of 2 network server statistics across several machines: the latency and throughput of each machine. You will be using ex8.py for this section of the exercise.

To perform anomaly detection, we will first need to fit a model to the data's distribution. We want to estimate the Gaussian distribution for each of the features across training data. Note that the Gaussian distribution for a certain feature of a certain dataset can be represented by its mean and variance. Here, estimateGaussian.py estimates the mean and variance of the data. After the function call to estimateGaussian() is finished, the next part of ex8.py will visualize the contours of the fitted Gaussian distribution.

Now, we want to find out which samples are anomalies. To do this, we need to select a threshold value based on a cross validation set. If a sample has a probability value lower than the threshold value, then it is considered to be an anomaly. Here, selectThreshold.py selects the threshold value (in variable epsilon) using the F1 score on a cross validation set. It iterates over a loop, computing the F1 score of a chosen epsilon value as the threshold in each iteration. At the end of the loop, it will check if the F1 score for the current epsilon choice is greater than the highest F1 score ever computed among the previous choices of epsilon, and if so, then it will update the best F1 score and best epsilon value. After the function call to selectThreshold() is finished, the next part of ex8.py will circle the anomalies in the figure shown from this exercise.

Your task is to complete the functions in ex8modules.py. Implement the two functions by filling in "YOUR CODE HERE" sections. After you have finished filling in your code, activate Miniconda, change directory to where your ex8.py is located, then type in following command and press Enter:

```
python ex8.py
```

During the execution, you will see output text results in console and graphic results in a separate window. If your implementation is correct, the graphic results will be similar to what is shown on the "Sample Results" section of this instructions sheet.

[Section 2: Recommender Systems]

In this exercise, you will implement the collaborative filtering learning algorithm and apply it to a dataset of movie ratings. You will be using ex8_cofi.py for this section of the exercise. ex8_movies.mat contains data on the movie ratings. It provides the variables Y and R in your Python environment. The matrix Y (a num_movies * num_users matrix) stores the ratings (from 1 to 5) of each movie rated by each users. The matrix R is an binary-valued indicator matrix, where R(i, j) = 1 if user j gave a rating to movie i, and R(i, j) = 0 otherwise. The objective of collaborative filtering is to predict movie ratings for the movies that users have not yet rated, that is, the entries with R(i, j) = 0. This will allow us to recommend the movies with the highest predicted ratings to the user.

Here, cofiCostFunc.py returns the cost value and gradient value for the collaborative filtering problem. ex8_cofi.py will call cofiCostFunc() to check your implementation of cost function value compuation for collaborative filtering. Next, the script ex8_cofi.py will run a gradient check (checkCostFunction) to numerically check the implementation of your gradients. It will do these on both non-regularized and regularized cases. Then, the script ex8_cofi.py will call scipy.optimize.minimize() to take the cofiCostFunc() function instance as an argument to learn the parameters for collaborative filtering. If your implementation is correct, the final recommendation results will be similar to as follows:

```
Top recommendations for you:

Predicting rating 7.92633 for movie Star Wars (1977)
Predicting rating 7.79929 for movie Titanic (1997)
Predicting rating 7.78432 for movie Schindler's List (1993)
Predicting rating 7.74309 for movie Shawshank Redemption, The (1994)
Predicting rating 7.60663 for movie Usual Suspects, The (1995)
Predicting rating 7.59832 for movie Good Will Hunting (1997)
Predicting rating 7.56825 for movie Raiders of the Lost Ark (1981)
Predicting rating 7.52306 for movie Close Shave, A (1995)
Predicting rating 7.50462 for movie Godfather, The (1972)
Predicting rating 7.48261 for movie Casablanca (1942)
```

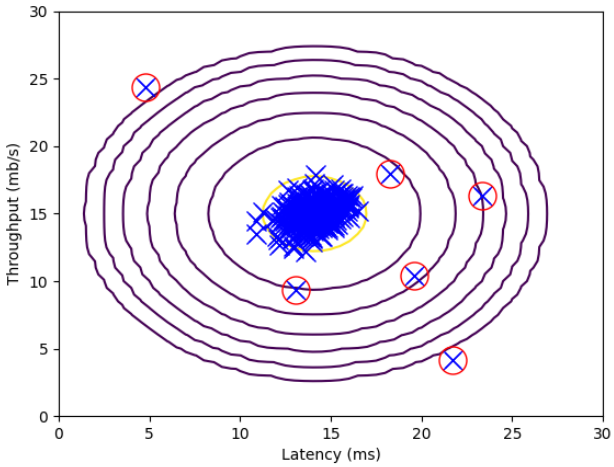
Your task is to complete the function in ex8modules_cofi.py. Implement the function by filling in “YOUR CODE HERE” sections. After you have finished filling in your code, activate Miniconda, change directory to where your ex8_cofi.py is located, then type in following command and press Enter:

```
python ex8_cofi.py
```

During the execution, you will see output text results in console and graphic results in a separate window. If your implementation is correct, the graphic results will be similar to what is shown on the “Sample Results” section of this instructions sheet.

Sample Results:

[Section 1: Anomaly detection]



[Section 2: Recommender Systems]