CIS 419/519 Introduction to Machine Learning
Assignment 5

Due: December 4, 2017 11:59pm

Instructions

Read all instructions in this section thoroughly.

Collaboration: Make certain that you understand the course collaboration policy, described on the course website. You must complete this assignment **individually**; you are **not** allowed to collaborate with anyone else. You may *discuss* the homework to understand the problems and the mathematics behind the various learning algorithms, but **you are not allowed to share problem solutions or your code with any other students.** You must also not consult code on the internet that is directly related to the programming exercise. We will be using automatic checking software to detect academic dishonesty, so please don't do it.

You are also prohibited from posting any part of your solution to the internet, even after the course is complete. Similarly, please don't post this PDF file or the homework skeleton code to the internet.

Formatting: This assignment consists of two parts: a problem set and program exercises.

For the problem set, you must write up your solutions electronically and submit it as a single PDF document. We will not accept handwritten or paper copies of the homework. Your problem set solutions must use proper mathematical formatting. For this reason, we **strongly** encourage you to write up your responses using LaTeX. (Alternative word processors, such as MS Word, produce very poorly formatted mathematics.)

Your solutions to the programming exercises must be implemented in python, following the precise instructions included in Part 2. Portions of the programing exercise will be graded automatically, so it is imperative that your code follows the specified API. A few parts of the programming exercise asks you to create plots or describe results; these should be included in the same PDF document that you create for the problem set.

Homework Template and Files to Get You Started: There is no homework template for this assignment. You can download the README from the course website, and the files for the RL problem from the RL problem's website.

Citing Your Sources: Any sources of help that you consult while completing this assignment (other students, textbooks, websites, etc.) ***MUST*** be noted in the your `README` file. This includes anyone you briefly discussed the homework with. If you received help from the following sources, you do not need to cite it: course instructor, course teaching assistants, course lecture notes, course textbooks or other readings.

Submitting Your Solution: We will post instructions for submitting your solution one week before the assignment is due. Be sure to check Piazza then for details.

CIS 519 ONLY Problems: Several problems are marked as "[CIS 519 ONLY]" in this assignment. Only students enrolled in CIS 519 are required to complete these problems. However, we do encourage students in CIS 419 to read through these problems, although you are not required to complete them.

All homeworks will receive a percentage grade, but CIS 519 homeworks will be graded out of a different total number of points than CIS 419 homeworks. Students in CIS 419 choosing to complete CIS 519 ONLY exercises will not receive any credit for answers to these questions (i.e., they will not count as extra credit nor will they compensate for points lost on other problems).

Acknowledgements: The RL problem has been adapted from materials used at UCBerkeley.

Assignment Version 20171110a

PART I: PROBLEM SET

Your solutions to the problems will be submitted as a single PDF document. Be certain that your problems are well-numbered and that it is clear what your answers are. Additionally, you will be required to duplicate your answers to particular problems in the `README` file that you will submit.

1 K-Means (10 pts)

Show 2 iterations of the k-means algorithm ($k = 2$) on the following one-dimensional data set:

Data: [4, 1, 9, 12, 6, 10, 2, 3, 9]

First iteration: cluster centers (randomly chosen): 1, 6

Data assignment:

- Cluster 1: { 1, 2, 3 }
- Cluster 2: { 4, 9, 12, 6, 10, 9 }

(a) Show the cluster centers, then the data assignments, that would be obtained for each of two more iterations.

(b) After your iterations, has the algorithm converged to a solution at this point, or not? How can you tell?

2 K-Means and Variance (6 pts)

(a) When using the K-Means clustering algorithm, we seek to minimize the variance of the solution. In general, what happens to the variance of a partition as you increase the value of K (the number of clusters) and why? State your answer in one sentence.

(b) For a dataset with n instances, what value of k can you always get a variance of 0? Why? State your answer in one sentence.

3 Reinforcement Learning I (10 pts)

(Adapted from Sutton and Barto, Ex. 3.5) Imagine that you are designing a robot to run a maze. You decide to give it a reward of +1 for escaping the maze and a reward of zero at all other times. The task seems to break down naturally into episodes – the successive runs through the maze – so you decide to treat it as an episodic task, where the goal is to maximize expected total reward $R_t = r_{t+1} + r_{t+2} + r_{t+3} + \ldots + r_T$, where T is the final time step of an episode. After running the learning agent for a while, you find that it is showing no improvement in escaping from the maze. Something is going wrong.

Does the reward function effectively communicate the goal to the agent? If not, can you suggest another reward function that will work? If the reward function is fine, what else is going wrong?

4 Reinforcement Learning II (CIS 519 ONLY – 15 pts)

(Adapted from Sutton and Barto, Ex. 3.10.) Consider reinforcement learning in a gridworld where rewards are positive for goals, negative for running into the edge of the world, and zero otherwise.

(a) Are the signs of these rewards important, or only the intervals between them?

(b) Provide a formal proof that adding a constant C to all the rewards simply adds a constant, K, to the values of all states, and thus does not affect the relative value of any policies. In your proof, you will find it useful to use the following equations we studied in class for the expected discounted return and value of a state:

$$R_t = \sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \qquad\qquad V^\pi(s) = \mathbb{E}_\pi\Big[R_t \mid s_t = s\Big]$$

(c) What is K in terms of C and gamma?

PART II: PROGRAMMING EXERCISES

1 Image Segmentation using K-Means (30 pts)

In this problem, you will apply K-Means to image segmentation. Write a program named `imageSegmentation.py` that reads in an image, segments that image using K-Means clustering as described below, and outputs the new segmented image. Your program must support the following command line arguments:

```
python imageSegmentation.py K inputImageFilename outputImageFilename
```

The first argument `K` is an integer greater than 2 that specifies the number of clusters, `inputImageFilename` is the filename of the input image, and `outputImageFilename` is the filename to write the output image. For example, we might call your program via:

```
python imageSegmentation.py 24 newyorkcity.jpg nyc-segmented.jpg
```

Choose several nice natural images, such as a farmhouse against a blue sky, or a city scene. First write code to load the image using the Python Image Library (you might find it useful to consult http://en.wikibooks.org/wiki/Python_Imaging_Library). The Python Image Library supports a wide variety of image file formats, and will automatically determine the filetype based on the file extension. (We will test your program with .jpg and .png files, so make certain to test your program with those types.)

We can think of an image as being represented as a 3-D matrix of size $imageWidth \times imageHeight \times 3$. For each location in the image (i, j), the matrix contains three values for the red, green, and blue components of the pixels. We will use these pixel values for clustering. In addition to the color values (r_p, g_p, b_p) for pixel p, we will also use the x,y coordinates (i_p, j_p) as features. In particular, we can represent each pixel p as a five-dimensional data vector $\mathbf{x}_p = \begin{bmatrix} r_p & g_p & b_p & i_p & j_p \end{bmatrix}$.

Complete the program via the following steps:

- Convert the input image into a data set with five features, as described above. To improve results, you should also standardize the values of each feature in the data set.
- Implement your own version of K-Means and use it to cluster the data (i.e. the features for each pixel) into `K` clusters. If a cluster is ever empty during the procedure, assign a random data point to it. Use random initializations for the cluster centers, and iterate until the centroids converge.
- Use the cluster centers to generate the segmented image by replacing each data point's color values with the closest center. For example, $\mathbf{x}_p$ becomes
$$\hat{\mathbf{x}}_p = \begin{bmatrix} r_{\mathcal{C}(p)} & g_{\mathcal{C}(p)} & b_{\mathcal{C}(p)} & i_p & j_p \end{bmatrix} ,$$
where $\mathcal{C}(p)$ is the cluster to which $\mathbf{x}_p$ belongs and $(r_{\mathcal{C}(p)}, g_{\mathcal{C}(p)}, b_{\mathcal{C}(p)})$ are the corresponding RGB values of that cluster's centroid. Note specifically that we're only replacing the color values of each instance with its centroid's colors, we're **not** changing the (i,j) coordinates of that instance.
- Create an output image the same size as the input image. Then fill in the color values of each pixel of the image based on the $\hat{\mathbf{x}}_p$'s. For example, $\hat{\mathbf{x}}_p$ informs us that the pixel at (i_p, j_p) should have color $(r_{\mathcal{C}(p)}, g_{\mathcal{C}(p)}, b_{\mathcal{C}(p)})$. Note that you also have to undo the feature standardization at this point (just invert the standardization equation by solving for the original value given the standardized value).
- Output the resulting image to the file `outputImageFilename`.
- In your PDF writeup, include three different examples of an original image alongside the resulting segmented image.

The result of this process is called an over-segmented image. It is the first step to building such systems as this: http://make3d.cs.cornell.edu/. Later steps would piece these segments together into objects.

2 Reinforcement Learning (44 pts for CIS 419; 52 pts for CIS 519)

The reinforcement learning part of this assignment is rather detailed, and has its own webpage: https://www.cis.upenn.edu/~cis519/fall2017/homework/hw5reinforcement/reinforcement.html.