Assignment 3: Unsupervised Learning

Submit Assignment

Due Tuesday by 9:30am **Points** 100 **Submitting** a file upload
Available Nov 6 at 12am - Nov 20 at 9:45am 14 days

Assignment 3: Unsupervised Learning

Released: 11/06/18
Due: 11/20/18, beginning of class

The assignment is worth 10% of your final grade.

Collaboration: You must complete and turn in this assignment individually. The “whiteboard policy" is in effect. (Please see the collaboration policy on the course website for details).

Assignment Instructions

It is time to explore unsupervised learning algorithms. This part of the assignment asks you to use some of the clustering and dimensionality reduction algorithms we've looked at in class and to revisit earlier assignments. The goal is for you to think about how these algorithms are the same as, different from, and interact with your earlier work.

The same ground rules apply for programming languages.

Read everything below carefully!

The Problems Given to You

You are to implement (or find the code for) three algorithms. The first two are clustering algorithms:

- **k**-means clustering
- GMMs via Expectation Maximization

You can choose your own measures of distance/similarity. Naturally, you'll have to justify your choices, but you're practiced at that sort of thing by now.

The last algorithm is a dimensionality reduction algorithm:

- PCA

You are to run a number of experiments. Come up with at least two datasets. If you'd like (and it makes a lot of sense in this case) you can use the ones you used in the first and/or second assignment.

1. Run the clustering algorithms on the data sets and describe what you see.
2. Apply PCA to the two datasets and describe what you see.
3. Reproduce your clustering experiments, but on the data **after** you've run dimensionality reduction (PCA) on it.

4. Apply PCA to one of the datasets from assignment #1 (you may have used this dataset for experiments 1-3 above), by learning a projection solely from the training set and projecting both the training and test sets. Rerun your neural network learner on the newly projected data.
5. Apply the two clustering algorithms to the same dataset to which you just applied PCA, treating the clusters as if they were new features. For example, for GMM, the probability of belonging to each cluster becomes new features. In other words, run the clustering algorithms on the raw data, but treat them as if they were dimensionality reduction algorithms. Note that as in 4 above, this dimensionality reduction projection is learned from the training set only, but it's used to project both training and test sets. Again, rerun your neural network learner on the newly projected data.

In summary: you should perform 6 experiments where you run and analyze the performance of the 3 unsupervised learning algorithms (k-means, GMMs, and PCA) on two datasets. You should do 4 more experiments where you run k-means and GMMs on the two datasets after doing PCA. 1 experiment where you first apply PCA, then train the neural network on the projected data. 2 experiments where you sue clustering as dimensionality reduction and retrain the neural network on the reduced data. This is a grand total of 13 experiments.

What to Turn In

You must submit a tar or zip file named **yourgtaccount**.{zip,tar,tar.gz} in t-square that contains a single folder or directory named **yourgtaccount** that in turn contains: -->

1. A file named **README.txt** that contains instructions for running your code
2. your code
3. a file named yourgtaccount-**analysis.pdf** that contains your writeup.
4. any supporting files you need (for example, your datasets).

The file yourgtaccount-**analysis**.pdf should contain:

- a discussion of your datasets, and why they're interesting: If you're using the same datasets as before at least briefly remind us of what they are so we don't have to revisit your old assignment write-up.
- explanations of your methods: How did you choose **k**?
- a description of the kind of clusters that you got.
- analyses of your results. Why did you get the clusters you did? Do they make "sense"? If you used data that already had labels (for example data from a classification problem from assignment #1) did the clusters line up with the labels? Do they otherwise line up naturally? Why or why not? Compare and contrast the different algorithms. What sort of changes might you make to each of those algorithms to improve performance? How much performance was due to the problems you chose? Be creative and think of as many questions you can, and as many answers as you can. Take care to justify your analysis with data explicitly.
- Can you describe how the data look in the new spaces you created with the various algorithms? For PCA, what is the distribution of eigenvalues? Assuming you only generate **k** projections (**i.e.**, you do dimensionality reduction), how well is the data reconstructed by PCA?
- When you reproduced your clustering experiments on the datasets projected onto the new spaces created by PCA, did you get the same clusters as before? Different clusters? Why? Why not?
- When you re-ran your neural network algorithms were there any differences in performance? Speed? Anything at all?

It might be difficult to generate the same kinds of graphs for this part of the assignment as you did before; however, you should come up with some way to describe the kinds of clusters you get. If you can do that visually all the better.

Note: Analysis writeup is limited to 10 pages total.

Grading Criteria

At this point you are not surprised to read that you are being graded on your analysis more than anything else. I will refer you to this section from assignment #1 for a more detailed explanation. As always, start now.