**Why?**

Now it's 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) six algorithms. The first two are clustering algorithms:

* *k*-means clustering
* 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 four algorithms are dimensionality reduction algorithms:

* PCA
* ICA
* Randomized Projections
* Any other feature selection algorithm you desire

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 assignment.

1. Run the clustering algorithms on the data sets and describe what you see.
2. Apply the dimensionality reduction algorithms to the two datasets and describe what you see.
3. Reproduce your clustering experiments, but on the data after you've run dimensionality reduction on it.
4. Apply the dimensionality reduction algorithms to one of your datasets from assignment #1 (if you've reused the datasets from assignment #1 to do experiments 1-3 above then you've already done this) and rerun your neural network learner on the newly projected data.
5. Apply the clustering algorithms to the same dataset to which you just applied the dimensionality reduction algorithms (you've probably already done this), treating the clusters as if they were new features. In other words, treat the clustering algorithms as if they were dimensionality reduction algorithms. Again, rerun your neural network learner on the newly projected data.

**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 explictly.
* Can you describe how the data look in the new spaces you created with the various aglorithms? For PCA, what is the distribution of eigenvalues? For ICA, how kurtotic are the distributions? Do the projection axes for ICA seem to capture anything "meaningful"? Assuming you only generate *k* projections (*i.e.*, you do dimensionality reduction), how well is the data reconstructed by the randomized projections? PCA? How much variation did you get when you re-ran your RP several times (I know I don't have to mention that you might want to run RP many times to see what happens, but I hope you forgive me)?
* When you reproduced your clustering experiments on the datasets projected onto the new spaces created by ICA, PCA and RP, 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.

**Submitted Attachments**

* File attachment [vsunkara6.zip](https://t-square.gatech.edu/access/content/attachment/gtc-366b-12ae-5092-b5c5-8db1fba74f17/Assignments/2d0ffb34-835a-4456-871c-8507cd6524a2/vsunkara6.zip) ( 1 MB; Nov 6, 2016 10:21 pm )

**Additional instructor's comments about your submission**

 Overall Feedback:

- I have no idea what your first histogram plot is? Even if I did I see no title, axis labels, and they are all way too small

Clustering and dimension Reduction:

- “Below are the scatter plots for both data sets using kmeans and EM clustering algorithms, run on weka tool. Instructions are mentioned in the README file. These plots are generated by first applying PCA and then keeping only first four eigen vectors and reducing the data sets to 2D “ <- I’m extremely confused as to what you are doing here. Why are you already talking about PCA? You’re first task is to do clustering on the raw data. Why visualize in PCA space if you’re asked to run k-means and EM on the raw data?

- We want to see a more sophisticated analysis for how you picked k. Did you pick k to be the same as the number of class labels? -> “Hence, I selected 3 clusters for Abalone and 2 clusters for Wine Quality. “. Trying looking into the elbow method or other methods for picking the optimal number of clusters

- “A lower DBIndex value means a good cluster. Using this metrics also, EM wins over KMeans.” <- Do you think these results would have been the same if you had varied K?

- I don't see any interpretation of your clusters. What do your centers look like?

- Did you run your clustering algorithms with multiple restarts or just a single iteration? Was there variation for multiple restarts?

- Your pca vector dumps are hard to read. Make a table!!

- I’d like to see some analysis of the reduced dimensions. Can you visualize the two dimensions and see linear separability for the classes? What does the new projection look like for each of the dimension reduction algorithms?

- Is there any interpretation of the reduced dimensions? What are the loadings of the different variables on the 1st PCA dimension for example?

- Okay so you clustered on the reduced dimensions and calculated a bunch of metrics, but I don’t see any analysis. What do the new clusters tell you?

“We’ve a parameter to set the number of independent components. Keep it as -1 will take total number of independent features as the components. I’ve set this value as 3. “ <- I can’t understand a lot of this sentence, but how did you decide on 3? Did you look at kurtosis at all?

- “I’ve picked bottom 4 based on a threshold value.  <- Explain in more detail. What’s the point of the threshold value?

- I’d like to see more explanation about how to picked the number of Randomized projections

Neural Networks:

- I have no idea what your x and y labels are for your neural network graphs. I’m assuming it is training episodes vs. error

- What about training time for all your Neural Network runs?

- Why do you think PCA helped reduce the error rate?

-“Abalone Cluster Projections “ <- you only had to run one dataset through your neural network, but I have no idea what you actually ran? What clustering output did you hand to your neural network? K-means, em on raw data? K-means, em on PCA reductions?

Top of Form

Bottom of Form