HW 7 Example Answers

I’ve pulled answers from your homeworks that I thought were particularly good examples. Obviously there’s no single correct answer to these questions.

**1. Using an MST to infer developmental trajectories**

a. No. Given that the edge weights represent expression difference distances, to find the most likely path would be equivalent to finding the shortest path of the given graph. However, an MST does not necessarily produce the shortest path between any two vertices in a graph. (MSTs just produce a tree with the lowest weight). Thus, we would instead prefer an alternative algorithm such as Dijkstra’s which does in fact find the shortest path.

b. Yes, a minimum spanning tree is a good estimate to use for inferring the relationships between the cell types. This is because we believe that the neurons are developmentally related, so we can infer that one type may have developed from another type. As a result, the neurons may have developed from a similar cell ancestor, and their development may have split off later on. We can assume that the development ancestor may be somewhat represented in the cells by an existing cell that we have sampled. If not this is ok, as the MST approach should still work, but maybe not as well. The MST approach will capture the ancestor/child relationships between different neurons because it will find the way with the smallest overall path length to represent the neurons, and this should capture developmental relationships, because if one neuron development is a little farther than another's, it should show in the MST as an edge between the neurons, as one represents a previous state and one represents a little more developed state. Using the MST is better than a pairwise comparison of distances, as the MST represents the development of the neurons, and some that may be relatively close in expression may be far away developmentally and may have just happened to end in similar gene expression patterns. As a result, the MST approach is good, and it is also used in other programs such as Monocle1, so this is an approach that has been used in the literature.

**c. Some thoughts: There are a couple of data considerations that impact the approaches. As several people observed, the MST provides a single tree, which might eliminate important relationships between cells. MST would work well if we do have cells that represent the developmental ancestral state, as it would generally choose those paths as the best . Alternatively, dikstra’s is good here, but this will work well only if we have a sparsely connected network (or a degenerate one) that allows us to find a connecting path between two cells that isn’t just the edge between 2 cells. Our initial network has edge weights between every node, so it is likely Dikstra’s would just give us the known distance between cells. We could use some sort of low-pass filtration for the edge weights to improve this. Even if this is not the case, it is perhaps not an ideal approach if we want to describe all of the data (since each path would be between just two nodes). If ancestral cells aren’t represented in the data, the Steiner tree might provide a good answer, but comes with additional computational complexity issues.**

**2. Accoustic similarity of words as the feature space**

a. If the goal is to compare word choice, relating vertexes by words that sound similar seems like a bad choice. For example, cat and cut would be closely connected even though they describe completely different ideas and would not be used to describe the same movie regardless of sleep deprivation. A better way to measure edge weights would be based on word definition similarity. Perhaps by measuring the steps required to get between two words in a thesaurus would give a better measure of edge weights.

b. This is not a good design of the feature space and edge weights. Yongjun is interested in word choice after sleep deprivation. Therefore, it is not useful to weigh edges by acoustic similarity; this does not add information about the problem Yongjun hopes to answer. A more reasonable approach is to define a complete graph in which each node is word. Then, pairwise weight between nodes is the number of times those words are spoken in succession by any of the subjects. In this way, there is a path through the graph describing a subject's speech and the sum of the path is measure of "word choice". Smaller sums describe a more unique choice and ordering of words.

**c. Thoughts: A couple people had some good ideas for how to get some mileage out of the approach we proposed, but overall, it’s just not the right approach for this question. There are many ways to define the graph that work better.**

**3. tSNE to compare variability of facebook data**

a. Two facts aboutt-SNE that may be relevant here:

•The t-SNE mapping requires a distance calculation to define the starting probability distributions.

•Distances in the t-SNE output space are strongly affected by parameter choices, and thus cannot be directly interpreted (although in general, similar points will be mapped to similar areas in the output space).

T-SNE may be useful as a dimensionality reduction method in this case, but (following from the first fact) it would be necessary to choose a distance measure that makes sense on binary data. Euclidean distance could work here, but as the data is sparse and high-dimensional, this may give strange results. A distance measure such as cosine similarity may work better for high-dimensional data (this can easily be converted to a distance as 1 - similarity; cosine similarity also does not obey the triangle inequality but that likely wouldn’t be an issue here, indeed cosine similarity is commonly used as a distance measure in t-SNE) However, it does not make sense to cluster the data after using t-SNE for dimensionality reduction (following from the second fact). Clustering the data (using, e.g. k-means clustering) in the higher-dimensional space, then subsequently visualizing it using t-SNE, would make the most sense.

b. I do not think the current approach is the best way to approach the problem. First of all, the Facebook data will have a ridiculous amount of information and dimension, potentially in the billions. Each person's interests will likely have very little overlap, and as a result it is not going to give much information to represent a billion dimension data set in two dimensions, as two dimensions cannot represent it well! The clustering is a good idea potentially, and a better approach may be to first group the interests into categories, such as different types of music etc. in order to reduce the dimension by a lot, and then perform clustering (without t-SNE). If this is too computationally hard, then we can use t-SNE, but it is important to group interests into categories, because they are related to each other. Another approach may be to use a graph with each interest, and to use edge weights between interests to represent the relationship among interests. Closer interests will be represented closer in the graph. Then where each student's interests are in the graph could be used to study where each student lies in the interest space, which could be reduced then to 2 dimensions using t-SNE. A final approach is to use a machine learning method to classify students as one of each type in terms of interests, but this would require labeling some students as already having an interest type. Finally, one thing to mention is that people may not provide racial information on Facebook, so some types of diversity cannot be determined by interests alone (but maybe a machine learning algorithm on images could determine this).

**c. Thoughts: This was tough as we hadn’t discussed t-SNE yet, but many of you gave very thoughtful answers on why t-SNE was or was not appropriate, given some considerations. One very important point: many of you explained why t-SNE was good/bad but did not compare it to some other method. If this were a question on the final, and you left out a part like that, you would lose partial credit. Make sure you answer all portions of the question on these longer form prompts.**