CS 6966: High-Dimensional Data Analysis **Due: Thursday 10.31 at 11:50pm**

Homework 3: Dimensionality Reduction and Similarity Search

Instructions: Your answers are due at 11:50pm submitted on canvas. You must turn in a pdf through canvas. I recommend using latex (`http://www.cs.utah.edu/~jeffp/teaching/latex/`, see also `http://overleaf.com`) for producing the assignment answers. If the answers are too hard to read you will lose points, entire questions may be given a 0 (e.g. **sloppy pictures with your phone's camera are not ok, but very careful ones are**)

Please make sure your name appears at the top of the page.

You may discuss the concepts with your classmates, but write up the answers entirely on your own. **Be sure to show all the work involved in deriving your answers! If you just give a final answer without explanation, you may not receive credit for that question.**

---

1. **[40 points]** Consider this data set [`https://users.cs.utah.edu/~jeffp/teaching/HDDA/homeworks/glove_word_embeddings.json`] which contains 7 sets of 10 words each. Each set has one word type { nouns, pronouns, verbs, prepositions, adverbs, adjectives, and conjunctions} in 100 dimensions. Use each of

   - PCA (Principal Component Analysis)
   - t-SNE (t-distributed Stochastic Neighborhood Embedding)
   - LDA : Linear Discriminant Analysis

   to project the words to 2 dimensions.

   Given the resulting plots, discuss the pros and cons of each method on this data set.

2. **[20 points]** For Johnson-Lindenstrauss Random Projections, we need $m = O(\frac{1}{\varepsilon}\log(n/\delta))$ target dimensions to achieve $(1+\varepsilon)$-distortion on n distances, with probability $1-\delta$. For the purposes of this question, lets assume the exact ratio is $m = 2\frac{1}{\varepsilon}\ln(n/\delta))$ where ln is the natural log (based e).

   Use this formula to calculate the following values based on the input parameters:

   (a) $n = 1,000$ data points, $\varepsilon = 0.1$ error and $\delta = 0.05$ probability of failure. How many target dimensions m do you need?

   (b) $n = 100,000$ data points, $\varepsilon = 0.1$ error and $\delta = 0.05$ probability of failure. How many target dimensions m do you need?

   (c) $n = 10,000$ data points, $\delta = 0.1$ probability of failure, and $m = 1000$ target dimensions. How much distortion $(1+\varepsilon)$ can you guarantee?

   (d) $n = 10,000$ data points, $\delta = 0.1$ probability of failure, and $m = 10,000$ target dimensions. How much distortion $(1+\varepsilon)$ can you guarantee?

3. **[40 points]** Download or import a pre-trained word embedding model, such as GloVE (`https://nlp.stanford.edu/projects/glove/`) or word2vec of dimension at least 50, and

at least 100,000 words. Load the dataset into a fast (approximate) nearest neighbor search software like FAISS, KGraph, or FALCONN. Compare the time to compute a nearest neighbor with the software with the runtime of brute force (compare distance to all other points). You should average your comparison over enough trials to be confident in your reporting.