Homework 2:
Similarity search and dimension reduction

Problem 1: Similarity metrics [15 points]

In this problem, you will work with various similarity metrics for data, and think about their pros and cons. We will be working with the well-known "20 newsgroups" dataset. Each article in the dataset belongs to a newsgroup (e.g., science, politics, etc.) and is represented as a "bag of words," a common way of representing textual data. There is a list $w_1, \ldots, w_N$ of all the words that occur in all the documents. To each document D, we associate a N-dimensional vector $\mathbf{x}$ whose i-th coordinate contains the number of occurrences of w_i in D.

This data is contained in `newsgroup_data.yaml` on the class webpage. This YAML file contains data in the format

```
newsgroup_name:
  article_id:
    word_id: count
    word_id: count
    word_id: count
    ...
  article_id:
    word_id: count
    word_id: count
    ...
```

Note that there are many unique words, i.e. N is very large, but any particular document will only contain a small set of them. **Therefore it will be important to only represent the non-zero entries of the bag-of-words vector.**

We'll work with the following similarity metrics, as covered in class. In the following, $\mathbf{x}$ and $\mathbf{y}$ are vectors representing two bags of words:

- Jaccard Similarity:

$$J(\mathbf{x},\mathbf{y}) = \frac{\sum_{i=1}^{N} \min(\mathbf{x}_i, \mathbf{y}_i)}{\sum_{i=1}^{N} \max(\mathbf{x}_i, \mathbf{y}_i)} .$$

- L_2 Similarity: $L_2(\mathbf{x},\mathbf{y}) = -\|\mathbf{x}-\mathbf{y}\|_2 = -\left(\sum_{i=1}^{n}(\mathbf{x}_i - \mathbf{y}_i)^2\right)^{1/2}$.

- Cosine Similarity:

$$S_C(\mathbf{x},\mathbf{y}) = \frac{\sum_{i=1}^{N} x_i \cdot y_i}{\|\mathbf{x}\|_2 \, \|\mathbf{y}\|_2} .$$

(a) [3 points] Implement the three similarity metrics described above. For each metric, prepare the following heatmap. The plot will be a 20×20 matrix. Rows and columns are indexed by newsgroups. For each entry (A, B) of the matrix, compute the average similarity of articles in group A with articles in group B, across all possible pairs of articles in these groups.

(b) [6 points] Based on the heatmaps you created, which of the similarity metrics seem the most suitable for the data, and why would you expect this to be the case? Are there any pairs of newsgroups that appear to be very similar? Can you explain why?

(c) [6 points] Imagine now that there was actually a secret twenty first newsgroup that was mistakenly mixed into the dataset, so that a 1/21-fraction of the data points labeled with each of the 20 newsgroups was actually from this other newsgroup. The structure of the documents in this hidden newsgroup could be arbitrary. How could this hidden newsgroup affect our heatmap? Explain which metrics appear to be more sensitive to the presence of this hidden newsgroup, and how the heatmap could change based on the structure of the documents in this hidden newsgroup.

Problem 2: Dimension reduction [20 points]

You may have noticed that it takes a non-trivial amount of time to compute pairwise distances for Problem 1 (on the order of several minutes). Here, we will explore how we can use dimension reduction to speed up this computation. Recall that our vectors in Problem 1 are N dimensional. Our goal will be to reduce them to a target dimension d, where $d \ll N$.

(a) [3 points] Implement a baseline cosine-similarity nearest-neighbor classification system that, for any given document, finds the document with largest cosine similarity and returns the corresponding newsgroup label. You should use brute-force search.

Compute the 20×20 matrix whose (A, B) entry is defined by the fraction of articles in group A that have their nearest neighbor in group B. Plot the results using a heat map as in Problem 1.

What is the average classification error (i.e., what fraction of the 1000 articles don't have the same newsgroup label as their closest neighbor)?

(b) [3 points] Suppose we're building an article classification system based on the algorithm you just implemented. We have a dataset of n labeled articles, a total of N distinct words across the articles, and m distinct newsgroups. What is the asymptotic (Big-Oh) runtime of the algorithm implemented in Part (a)?

Consider the following 2 types of random "sketching matrices" $M : \mathbb{R}^{d \times N}$.

- Case 1: Each entry of M is drawn randomly and independently from a Gaussian distribution with mean 0 and variance 1.
- Case 2: Each entry of M is drawn uniformly and independently from $\{-1, +1\}$.

Finally, also consider the following deterministic sketching matrix. When N is a power of 2, the *Walsh matrix of size* N is a $N \times N$ matrix defined recursively by

$$H_1 = \begin{bmatrix} 1 \end{bmatrix}, \qquad H_{2\ell} = \begin{bmatrix} H_\ell & H_\ell \\ H_\ell & -H_\ell \end{bmatrix} \quad (\ell \geq 1).$$

Now, let your sketching matrix be the first d rows of the the Walsh matrix of size N. Note that for us, N need not be a power of 2. In this case, just pad out each vector with 0's at the end until the length of the vector is a power of 2, and treat that as the length of the vector.

For each type of sketching matrix M, perform the following algorithm. Given a set of N-dimensional vectors $\mathbf{x}_1, \ldots, \mathbf{x}_n$, the d-dimensional reduced vector corresponding to $\mathbf{x}_i$ is given by the matrix-vector product $M\mathbf{x}_i$. One way to think about M is as a set of d random N dimensional vectors $\mathbf{w}_1, \ldots, \mathbf{w}_d$ (i.e. the rows of M), and the j-th coordinate of Mv_i is just $\langle \mathbf{v}_i, \mathbf{w}_j \rangle$.

(c) [4 points] Plot the nearest-neighbor visualization (heat map) as in part (a) for $d = 10, 30, 60, 120$, for each of the 3 types of matrices.

What is the average classification error for each d? For which values of the target dimension are the results comparable to the original data set?

(d) [5 points] How does the choice of the sketching matrix affect the classification error? Can you offer a plausible explanation for why?

(e) [3 points] Repeat the Big-Oh analysis for the new dimension-reduced algorithm, in terms of n, N, m and now d as well.

(f) [5 points] Why is it a good idea to do a randomized sketch? Show that there is a dataset of $N - d$ points where *any* deterministic sketch would be useless. Find such a data set for the deterministic sketch matrix described above. Offer a plausible explanation for the difference in behavior between the real data set and this constructed data set. **Hint: M has rank at most d.**

(g) [3 points] Why does randomizing the sketching matrix help avoid these issues?