University of Washington March 5, 2021
Department of Computer Science and Engineering
CSEP 521, Winter 2021

Homework 9, Due Thursday, March 11, 2021

Problem 1 (10 points):
This problem is to compute the Johnson Lindenstrauss (JL) transform on a small example. Consider the following points in R^4:

$$\begin{aligned} p_1 &= [-7, -4, 6, 2] \\ p_2 &= [2, -3, 1, 4] \\ p_3 &= [1, -10, 1, 3] \\ p_4 &= [-5, -4, -6, 5] \end{aligned}$$

a) Compute the Euclidean distances for all pairs of points.

b) Compute the projection of the points p_1, p_2, p_3 and p_4 onto the two vectors

$$b_1 = [-.18, -.72, .43, -.38] \text{ and } b_2 = [-.71, -1.23, .71, .60].$$

c) Compute the Euclidean distances (in R^2) for the projected points.

d) Compute the projection of the points p_1, p_2, p_3 and p_4 onto the two vectors

$$c_1 = [1, 1, -1, 1] \text{ and } c_2 = [1, -1, 1, 1].$$

e) Compute the Euclidean distances (in R^2) for the projected points.

Note: in the JL transform there is a scaling factor of $\frac{1}{\sqrt{d}}$ that is being ignored in this exercise.

Problem 2 (10 points) String edit distance for typo correction:
Given a pair of strings s_1 and s_2 the edit distance is the number of operations (such as adding, deleting or changing characters) to convert s_1 to s_2. The edit distance measure can be enhanced to give different costs for inserting or changing particular characters. For example it might“cost” one to change ‘a’ to ‘s’ and “cost” two to change ‘a’ to ‘d’. One motivation for this is to develop a model of most likely errors from typing in order to rank candidates in correction options.

a) Design a cost structure for edit operations that would be appropriate for modelling errors for typing on a mobile device while walking. (Obviously, there is a lot of flexibility in how you answer this. Provide some justification for your choices.)

b) Give the pseudo-code for computing the edit distance with your measure for a pair of strings. (For this problem, you are not required to reconstruct what the edit sequence is, just give the edit distance.)

Programming Problem 3 (20 points) Dimension reduction:

Goal: Explore the trade-off in dimension reduction between quality and number of dimensions.

Turn in: Code, figures, times, and discussion for parts (a)-(c). The programming problem will draw from Homework 8. You may reuse some of your code.

Given parameters n and d, define a $d \times n$ matrix M by drawing each entry of M randomly and independently from a normal distribution with mean 0 and variance 1.

Given a n-dimensional vector $v \in R^n$, you compute its image in d dimensions using the matrix-vector product Mv.

a) Implement the random projection dimension reduction method and plot the nearest-neighbor visualization as in Assignment 8, Problem 4d for cosine similarity and $d \in \{10, 25, 50, 100\}$. Also record the wall-clock time for each run. For which values of the target dimension are the results comparable to the original embedding?

b) Consider article 3 (from alt.atheism) and for each dimension $d \in \{10, 25, 50, 100\}$, create a scatter plot where each point represents an article, with the x-coordinate representing the cosine similarity to article 3 in the original (n-dimensional) space, and the y-coordinate representing the cosine similarity to article 3 in the d-dimensional space. You should compare article 3 with all other articles unless you run into performance or graphing difficulties, in which case, you can reduce the number of groups. Discuss the plots and how the error relates to the target dimension d.

c) Implement random projection by generating the matrix M in a different way: Generate M by choosing each entry to be an independent, uniformly random sign -1 or $+1$. Do part (b) again with this family of dimension reduction matrices (for $d \in \{10, 25, 50, 100\}$). Compare your results using the two methods.