Homework 4:
Principal component analysis

Problem 1: PCA vs least squares [21 points]

Both PCA and least squares regression (LS) are ways to infer linear relationships between data. In this problem, you will develop some intuition for the differences between these two approaches, and an understanding of the settings that are better suited to using PCA or better suited to using the least squares fit.

The high level bit is that PCA is useful when there is a set of latent (hidden/underlying) variables, and all the coordinates of your data are linear combinations (plus noise) of those variables. The least squares fit is useful when you have direct access to the independent variables, so any noisy coordinates are linear combinations (plus noise) of known variables.

In this problem we'll consider a simple synthetic example with two dependent variables x and y, where the hidden relationship between the two variables is that $y = 3x$, i.e. the goal is to recover the coefficient 3. As a reminder, given a dataset $(x_1, y_1), \ldots, (x_n, y_n)$ of predictors x_i and dependent values y_i, the least squares fit is the value that minimizes the least-squares error, i.e. least squares will return the line $\ell(x) : \mathbb{R} \to \mathbb{R}$ that minimizes

$$\sum_{i=1}^{n} (\ell(x_i) - y_i)^2 \ .$$

We'll be primarily interested in recovering the slope of the line; note that in 1D this has a very simple form, namely, if we let $x = (x_1, \ldots, x_n)$, and $y = (y_1, \ldots, y_n)$, then the slope of the minimizer is given by

$$\frac{\langle x - \mu_x \cdot \mathbf{1}, y - \mu_y \cdot \mathbf{1} \rangle}{\|x - \mu_x\|_2^2} \ ,$$

where μ_x and μ_y are the means of x and y, respectively, and $\mathbf{1}$ is the all-ones vector of length n.

(Warm-up / setup) [don't submit]

- Write a routine `pca_recover` that takes a vector $x = (x_1, \ldots, x_n)$ and a vector $y = (y_1, \ldots, y_n)$ and returns the slope of the first component of the PCA (namely, the second coordinate divided by the first).
- Write a routine `ls_recover` that takes x and y and returns the slope of the least squares fit.
- Set $x = (.001, .002, .003, ..., 1)$ and $y = 3x$. Make sure both routines return 3.

(a) [6 points] Let D be the distribution which is σ with probability 1/2 and $-\sigma$ with probability 1/2, so that the variance of D is σ^2. Suppose the x_i and y_i are all independent and drawn from D, for $i = 1, ..., n$. What does PCA recover, and what does LS recover, and how does the behavior change with n and σ for both? Briefly justify your answer.

(b) [3 points] We first consider the case where x is an independent variable that we get exactly, and we get noisy measurements of y. Let $x = (.001, .002, .003, ..., 1)$, and for a given noise level $\sigma > 0$, let

$$\widehat{y}_i \sim 3 \cdot x_i + \mathcal{N}(0, \sigma^2) = \frac{3i}{1000} + \mathcal{N}(0, \sigma^2) \ , \qquad i = 1, \ldots, 1000 \ .$$

Make a scatter plot with σ on the x-axis, and the output of `pca_recover` and `ls_recover` on the y-axis. For each $\sigma \in (0, 0.05, 1, \ldots, 0.45, 0.5)$, take a sample of $\widehat{y}_1, \ldots, \widehat{y}_{1000}$, plot the output of `pca_recover` as a red dot, and `ls_recover` as a blue dot. Repeat 40 times.

Note that in numpy, the code `numpy.randn(0, 1000)*s` generates an array of 1000 independent samples from $\mathcal{N}(0, s^2)$.

(c) [3 points] Now, we consider the case where there is noise on both x and y. For a given noise level $\sigma > 0$, and for $i = 1, \ldots, 1000$ let

$$\widehat{x}_i \sim x_i + \mathcal{N}(0, \sigma^2) = \frac{i}{1000} + \mathcal{N}(0, \sigma^2)$$
$$\widehat{y}_i \sim 3 \cdot x_i + \mathcal{N}(0, \sigma^2) = \frac{3i}{1000} + \mathcal{N}(0, \sigma^2) \ .$$

Just like in part (b), for each $\sigma \in (0, 0.05, 1, \ldots, 0.45, 0.5)$, take a sample of $\widehat{x}_1, \ldots, \widehat{x}_{1000}$ and $\widehat{y}_1, \ldots, \widehat{y}_{1000}$, plot the output of `pca_recover` as a red dot, and `ls_recover` as a blue dot. Repeat 40 times.

(d) [9 points] Answer the following:

1. Why does PCA do poorly with just noise on the y?
2. Why does PCA do well with noise on both x and y?
3. Why does LS do poorly with noise on both x and y?

Problem 2: PCA for genetic data [26 points]

The file `pca-data.txt` on the course webpage contains data from the 100 genomes project. Each of the lines in the file represents an individual. The first three columns contain: an individual's unique identifiers, his or her biological sex (1=male, 2=female), and the population to which they belong. The encodings for these populations can also be found in the course webpage. The subsequent columns of each line are a sample of the nucleobases from that individual's genome.

Convert the file into a matrix as follows. Let d be the number of columns, and n be the number of rows. For every column $j = 1, \ldots, d$, let η_j denote the most common nucleotide in that column. Call that the *mode* for column j. Construct an $n \times d$ matrix X as follows. For individual i, let $X_{ij} = 0$ if individual i has η_j in position j, and 1 otherwise. In other words, individual i has a 1 at column j if they have a mutation at position j.

Recall that if we are going to perform PCA on vectors $x_1, \ldots, x_n \in \mathbb{R}^d$, then we want to first *de-mean* then, i.e. replace them with $x_1 - \mu, x_2 - \mu, \ldots, x_n - \mu$, where

$$\mu = \frac{1}{n} \sum_{i=1}^{n} x_i \ .$$

In this problem, you will be asked to find the top k principal components, for different values of k. By this, we mean the k *orthonormal* vectors $v_1, \ldots, v_k$ that maximize the objective

$$\sum_{j=1}^{k} \sum_{i=1}^{n} \langle v_j, x_i \rangle^2 \ .$$

Recall, as discussed in lecture, that these vectors may not be unique, but they will be for this dataset. You may use the Python `sklearn` package to compute these components.

(a) [4 points] We will first examine the first 2 principal components of X. These components contain lots of information about our data set. Create a scatter plot with each of the 995 rows of X projected onto the first two principal components. In other words, the horizontal axis should be v_1 and the vertical axis v_2, and each individual should be projected onto the subspace spanned by v_1 and v_2. Your plot should use a different color for each population and include a legend. (Recall that the population data occurs in the third column.)

(b) [6 points] In two sentences, list 1 or 2 basic facts about the plot created in part (b). Can you interpret the first two principal components? What aspects of the data do the first two principal components capture? **Hint:** think about history and geography.

(c) [5 points] We will now examine the third principal component of X. Create another scatter plot with each individual projected onto the subspace spanned by the first and third principal components. After plotting, play with different labeling schemes (with labels derived from the meta-data) to explain the clusters that you see. Your plot must include a legend.

(d) [5 points] In one sentence, what information does the third principal component capture?

(e) [6 points] As noted previously, the top 3 principal components of X are all unique. How would you add an additional row to the matrix to make the first principal component *not* unique? For this additional row, you don't need to be restricted to a $\{0, 1\}$ vector. More generally, assume that the first $k + 1$ principal components are all unique. How would you add an row to make the k-th principal component not unique? Again, this row doesn't need to be $\{0, 1\}$-valued. **Hint: Use the fact that the principal components are orthonormal.**