Homework assignment #9-2

Due in class Mon 11/12

Getting comfortable with Principal Component Analysis (10 points)

In this problem we generate a high-dimensional dataset whose structure we know, and see if we can recover the signal directions using PCA. Use any programming environment you are comfortable with. The problem may seem long, but that's meant to make it easier, not harder.

Bob studies stem cells. He grows them in a lab, measures $D = 100$ characteristics he can think of, and hopes to infer something about their inner workings. Unbeknownst to Bob, the cells' behavior is entirely determined by the concentrations of just three key compounds: A, B and C. Thus, the state of a cell i is fully characterized by three numbers $\{a_i, b_i, c_i\}$. None of Bob's measurements probe these directly, but all the characteristics $\vec{x}_i$ that Bob is recording are related to these hidden variables in a simple linear way:

$$\vec{x}_i = \vec{x}_0 + a_i\,\vec{v}_1 + b_i\,\vec{v}_2 + c_i\,\vec{v}_3.$$

In the notation $\vec{x}_i$, the index i identifies the cell, and the vector notation refers to the hundred characteristics Bob measures, making each sample he collects a vector in a 100-dimensional space. The vectors $\vec{v}_1$, $\vec{v}_2$, $\vec{v}_3$ (also 100-dimensional) are constant and of **unit length**. Note that if characteristics do not change very much, this linear approximation is not even that silly – for small perturbations, all functions locally look like linear slopes.

The concentrations a_i, b_i and c_i vary from cell to cell: a_i is drawn from a Gaussian with **variance** $\lambda_a = 20$, b_i from a Gaussian with variance $\lambda_b = 5$, and c_i from a Gaussian of variance $\lambda_c = 0.5$. Separating the averages from the fluctuating component, we can write:

$$\vec{x}_i = [\vec{x}_0 + \langle a\rangle\vec{v}_1 + \langle b\rangle\vec{v}_2 + \langle c\rangle\vec{v}_3] + \delta a_i\,\vec{v}_1 + \delta b_i\vec{v}_2 + \delta c_i\vec{v}_3$$

$$\Rightarrow \vec{x}_i - \langle\vec{x}\rangle = \delta a_i\,\vec{v}_1 + \delta b_i\,\vec{v}_2 + \delta c_i\,\vec{v}_3$$

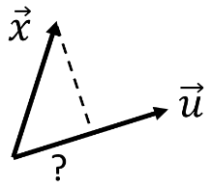
Here angular brackets denote averaging over samples i, and the δ's denote deviations from the mean. From now on, we can therefore thing about the coefficients a_i, b_i, c_i and the measured characteristics x_i as all having zero mean.

a. Generate three random vectors in D=100 dimensions: draw each component as a Gaussian random variable of mean 0 and variance 1, and normalize each vector to unit length. These will be your $\vec{v}_1, \vec{v}_2, \vec{v}_3$.
b. Calculate their scalar products and observe that they are almost orthogonal. This is another cool general feature of high-dimensional spaces!
c. Generate the random coefficients a_i, b_i, c_i describing the hidden states of $N = 500$ cells. Use them to generate the true values of all the characteristics x_i for these cells.
d. Bob's measurements are not perfect, but have a noise component: $x_{Bob} = x_{true} + \eta$. Let's take all η's to be Gaussian random variables of width $\sigma = 1$. In your code, generate the dataset as observed by Bob.

We now have a high-dimensional dataset where we know the ground truth – it looks 100-dimensional, but the underlying structure has dimension of just 3. Can we see this in our data?

e. For your dataset, make a scatter plot of the first characteristic against the second. Compute the variances of both these characteristics. Observe that both are close to σ^2.

f. Now instead of the first two dimensions, let's try two random directions. But first, a preliminary question: if $\vec{x}$ is some vector, and $\vec{u}$ is a unit vector, what is the length of the projection of $\vec{x}$ onto $\vec{u}$?



g. Generate two new random unit vectors as in (a). Project your data onto one and onto the other. Make a scatter plot of (projection onto first) vs. (projections onto second). Observe that again, the variances of both projections are basically $\sigma^2 \sim 1$.

Yet we know that somewhere in there is lurking a direction, the projection onto which has a huge variance ~20. How do we find it? Let's see if PCA can recover it!

h. Compute the D-by-D covariance matrix of the data: $C = \frac{1}{N} X^t X$. Diagonalize this matrix (MatLab, Python etc. all have the functionality of computing the eigensystem of a square matrix). Identify the first three eigenvectors.

i. Just as you did in (g), but now using the first eigenvector instead of a random direction: project your data onto the first eigenvector. What is the variance of this projection? Repeat for the second and third eigenvectors. Explain why one expects these variances to be close to $\lambda_a + \sigma^2$, $\lambda_b + \sigma^2$ and $\lambda_c + \sigma^2$, respectively. As in (g), make a scatter plot of the first two projections, and compare with the plot in (g).

j. Ideally, then, these eigenvectors should correspond to the true signal directions we "programmed" into our dataset. Compute the scalar product of the first eigenvector and $\vec{v}_1$, the second eigenvector and $\vec{v}_2$, the third eigenvector and $\vec{v}_3$. Explain why scalar products 1 and (-1) both correspond to perfect alignment/recovery. What do you observe? Has PCA recovered all three directions well?

k. Let's see if the answer in (j) is surprising, or was to be expected. The Marchenko-Pastur distribution predicts the eigenvalues of a pure-noise correlation matrix should follow:

$$p(x) = \frac{1}{2\pi\sigma^2} \frac{\sqrt{(\lambda_+ - x)(x - \lambda_-)}}{r\, x} \quad \text{for } x \in [\lambda_-, \lambda_+].$$

Here $r = D/N$ and $\lambda_\pm = \sigma^2\left(1 \pm \sqrt{r}\right)^2$. The expectation for the eigenvalues of C is the Marchenko-Pastur baseline from noise, plus three eigenvalues from signal (at $\lambda_a + \sigma^2$, $\lambda_b + \sigma^2$ and $\lambda_c + \sigma^2$). On the same plot, show both the Marchenko-Pastur distribution and the histogram of eigenvalues of your C. Comment on the answer you found in (j).

l. Inspired by the question (k), can you predict how many datapoints you would need to reliably recover the third (weakest) signal direction in this example (with $\lambda_c = 0.5$)?