Please provide complete and well-written solutions to the following exercises.

Due September 19, 4PM PST, to be uploaded as a single PDF document to brightspace. It is acceptable to instead upload a Jupyter Notebook, assuming you write in complete sentences where appropriate, and format your responses to be easily readable (i.e. if you only submit one big block of code with nothing written about what you did, then many points will be deducted from your score).

Homework 3

Exercise 1. In the Definition of PCA, we asserted that the diagonal matrix D can be assumed to satisfy $D_{ii} \geq D_{i+1,i+1}$ for all $1 \leq i \leq p-1$. Show that there always exists a singular value decomposition with this property. That is, if A is an $m \times n$ complex matrix with $m \leq n$, show that there exists an integer $p \leq m$, there exists a diagonal $p \times p$ matrix D with nonnegative values, there exist unitary $m \times m$ U and unitary $n \times n$ V such that

$$A = U \begin{pmatrix} D & 0 \\ 0 & 0 \end{pmatrix} V,$$

and such that $D_{ii} \geq D_{i+1,i+1}$ for all $1 \leq i \leq p-1$.

Exercise 2. Let A be a real $m \times n$ matrix, and define

$$||A||_{2\to 2} := \sup_{x \in \mathbf{R}^n : \, ||x||_2 \leq 1} ||Ax||_2 .$$

Let U be an $m \times m$ orthogonal matrix, let V be an $n \times n$ orthogonal matrix, let D be a $p \times p$ diagonal matrix with nonzero entries such that $D_{ii} \geq D_{i+1,i+1}$ for all $1 \leq i < p$. Show that

$$\left|\left| U \begin{pmatrix} D & 0 \\ 0 & 0 \end{pmatrix} V - \; U \begin{pmatrix} D & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} I_q & 0 \\ 0 & 0 \end{pmatrix} V \right|\right|_{2\to 2} = D_{q+1,q+1}.$$

So, if the singular value $D_{q+1,q+1}$ is small, then the original data matrix and the dimension reduced data matrix are not much different, in the sense that the $2 \to 2$ norm of their difference is small.

Exercise 3. Let $w^{(1)}, \ldots, w^{(m)} \in \mathbf{R}^q$. Let $y \in \mathbf{R}^q$. Show that

$$\sum_{j=1}^m \left\| w^{(j)} - \frac{1}{m} \sum_{\ell=1}^m w^{(\ell)} \right\|_2^2 \leq \sum_{j=1}^m \left\| w^{(j)} - y \right\|_2^2.$$

That is, the barycenter is the point in $\mathbf{R}^q$ that minimizes the sum of squared distances.

Exercise 4. Let $w^{(1)}, \ldots, w^{(m)} \in \mathbf{R}^q$ with $m \geq 2$. Show that

$$\sum_{i=1}^m \left\| w^{(i)} - \frac{1}{m} \sum_{j=1}^m w^{(j)} \right\|_2^2 > \sum_{i=1}^{m-1} \left\| w^{(i)} - \frac{1}{m-1} \sum_{j=1}^{m-1} w^{(j)} \right\|_2^2.$$

Exercise 5. In this exercise, we will perform some additional analysis on the sklearn built in data sets. For the iris data set, recall that we used PCA to embed the data points (i.e. 154 vectors in $\mathbf{R}^4$) into $\mathbf{R}^2$. We then ran `KMeans` from the sklearn package, with $k = 3$ clusters, and we plotted the resulting k clusters (with k different colors in a scatter plot), along with the k cluster centers

- Add some more information to our previous plot by also plotting the lines between the Voronoi regions. (Hint: if $z^{(1)}, z^{(2)} \in \mathbf{R}^2$ are two centers of two Voronoi regions, then the boundary between the regions is a line that is perpendicular to the straight line between $z^{(1)}$ and $z^{(2)}$, and this line passes through the point $(z^{(1)} + z^{(2)})/2$. This should be enough information to find the equation of this line.)
- Compute the percentage of mis-classified points (e.g. an output of 2% means that only about 2% of data points were mis-classified, i.e. about 98% of data points are correctly clustered)
- Sometimes a dataset is so large, it is difficult to directly use PCA on the data (e.g. the number 154 might be a trillion instead). We can still use PCA though by randomly sampling a small subset of rows of the data matrix, and hopefully performing PCA on this subset of the data will still be relevant for the full set of data. This statement can be made rigorous, but let's test it out ourselves. Randomly sample 20 rows from the data matrix (using e.g. the `random.sample` command from the `random` package), perform PCA on that resulting dataset, run k-means, and then repeat the above procedure to check for the number of mis-classified points (on the original dataset) that results from the clustering you got from the smaller dataset. (Once you have the cluster centers from the smaller dataset, you can then cluster the larger dataset using the cluster centers from the smaller dataset.) How did the number of mis-classified points change compared to performing PCA on your original dataset?

Repeat the above the for sklearn wine dataset (with $k = 3$), and the sklearn digits dataset (with $k = 10$). (Note: for the digits dataset, it might be hard to compute the exact percentage of mis-classified points; that is okay, just try to get a good estimate for the percentage.)

Exercise 6. Let A be an $m \times n$ matrix with nonnegative entries. A **nonnegative matrix factorization** for A with k classes is a factorization of the form

$$A = WH,$$

where W is an $m \times k$ matrix, H is a $k \times n$ matrix, and both W, H have nonnegative entries. Sometimes writing a factorization in this way is impossible. (If a factorization exists like this, then A must have rank at most k.) However, we can still try to find W, H that approximately satisfy $WH \approx A$. This is exactly what the Python function `NMF` does (from the `sklearn` package). (More specifically, Python tries to find W, H that minimize a norm of $A - WH$, plus some "regularizing terms". For details, see the sklearn documentation.)

Nonnegative matrix factorization is used in several machine learning applications, e.g. to cluster data into similar groups, in recommendation algorithms, etc. To illustrate this, let's

consider the matrix A whose entries are the numbers in the following table

	apple	banana	bell pepper	crab	broccoli	carrot	pear	shrimp
calories	130	110	25	100	45	30	100	100
sodium	0	0	40	330	80	60	0	240
potassium	260	450	220	300	460	250	190	220
carbohydrates	34	30	6	0	8	7	26	0
vitamin A	2	2	4	0	6	110	0	4
vitamin C	8	15	190	4	220	10	10	4

- Verify that A has rank 6, so that we know for sure we cannot write $A = WH$ exactly with $k = 3$.
- Find an approximate nonnegative matrix factorization of A with 3 classes with the Python commands

```
from sklearn.decomposition import NMF
model = NMF(n_components = 3, init = 'random', random_state = 0)
W = model.fit_transform(A)
H = model.components_
```

  Do the matrices W, H satisfy $A = WH$? If not, check the value of the norm of $A - WH$ and compare it with the norm of A.
- Each of the three rows of H corresponds to a different class of food. The largest entry in a column of H sorts the food into a given class. For example, the first row of H seems to correspond to "fruits," since the columns for apple, banana, and pear all have largest values in their top entries. (Carrot also seems to have a largest value here even though it is not a fruit.) What classes of foods do the other two rows of H seem to represent, and which food items are in those classes according to H?
- Each column of W also corresponds to a different class of food (like the rows of H). The largest entry in a row of W indicates which food characteristics are most important for being in each class. For example, calories, potassium, carbohydrates and vitamin A have their largest entries in the first column of W, so these four characteristics are the most significant contributions to being in the class of "fruits" in this table. (Carrot is the only one with a large value of vitamin A so it is unclear why exactly it got sorted in to the class of "fruits.") What food characteristics are most important for the other two classes of foods, according to W?
- When $k = 4$ instead of 3, is the carrot still in the same class as the apple, banana and pear?