Homework 4

Tejas Kamtam

305749402

Question 1 - PCA

Assuming the given matrices are not the "skinny" versions (i.e., if the dataset is not square there can be sections of the SVD matrices that result in all 0s which are usually chopped off - but in the answers below I assume this is NOT the case as the slides give definite shapes for these matrices - see slide deck 11 - PCA slide no. 22)

Part a

Yes - from the singular value matrix (D). The matrix is square with dimensions $n \times m$ where n is the number of samples (rows) and m is the number of variables (columns). So, there are 4 variables.

Part b

Given the assumption I made that this is not the skinny matrix, the resulted singular value matrix is $n \times m$ and the matrix is square so $n = m$, therefore we know there are 4 samples.

Part c

Yes - From the right-singular vector matrix, we know that each column represents a principal component direction/vector, so there are 4 PCs.

Part d

The explained variability ratio for some principal component PC_j is given as:

$$\text{Explained Variance}\big(PC_j\big) = \frac{d_j^2}{\sum_i d_i^2}$$

where d represents a singular value. Using this we get:

$$\text{Explained Variance}\big(PC_1\big) = \frac{30.6^2}{1361.2064} \approx 0.6879$$

$$\text{Explained Variance}\big(PC_2\big) = \frac{16.2^2}{1361.2064} \approx 0.1928$$

$$\text{Explained Variance}\big(PC_3\big) = \frac{11.2^2}{1361.2064} \approx 0.0922$$

$$\text{Explained Variance}\big(PC_4\big) = \frac{6.08^2}{1361.2064} \approx 0.0272$$

Part e

We can give linear combinations to calculate the first 2 PCs as (note we don't "re/un-transpose" as the given right-singular values are not indicative of the transposed matrix usually resulting from SVD):

$$PC_1 = S_{11} \cdot \Big(V_{1,1}x_1 + V_{1,2}x_2 + V_{1,3}x_3 + V_{1,4}x_4\Big)$$

$$PC_2 = S_{22} \cdot \Big(V_{2,1}x_1 + V_{2,2}x_2 + V_{2,3}x_3 + V_{2,4}x_4\Big)$$

Part f

Given the sample: $z = (0.65, -1.09, 1.21, 0.93)$, we can find the first 2 PCs using the formula from Part e. But, first, we can vectorize this to simplify calculation:

$$PC_1 = S_{11}\Big(V_1 \cdot z\Big) \approx 9.5350$$

$$PC_2 = S_{22}\left(V_2 \cdot z\right) \approx 15.5860$$

Question 2 - Clustering

Part a

We can create a table to better track distances:

Golfer	Drive	Fairway	D2Cindy	D2Charlotte	Class
Sung Ah	235.2	78.3	8.5	16.8	Cindy
Carin	236.8	67.8	10.1	15.2	Cindy
Erica	245.4	69.2	18.7	6.6	Charlotte
Charlotte	252.0	70.7	25.3	0	Charlotte
Cindy	226.7	72.1	0	25.3	Cindy

Re-calculating the centroids:

$$\text{Cindy Centroid} = (232.9, 72.73)$$

$$\text{Charlotte Centroid} = (248.7, 69.95)$$

Part b

We can model agglomerative clustering with iterative distance tables, such that every entry represents the distance from its row to its column. The first iteration is:

Golfer	Sung Ah	Carin	Erica	Charlotte	Cindy
Sung Ah	0	10.62	13.67	18.44	10.52
Carin	10.62	0	8.71	15.47	10.98
Erica	13.67	8.71	0	6.77	18.92
Charlotte	18.44	15.47	6.77	0	25.34
Cindy	10.52	10.98	18.92	25.34	0

From the table, we take the minimum, giving us the updated table using complete linkage:

Golfer	Sung Ah	Carin	Erica, Charlotte	Cindy
Sung Ah	0	10.62	18.44	**10.52**
Carin	10.62	0	15.47	10.98
Erica, Charlotte	18.44	15.47	0	25.34
Cindy	**10.52**	10.98	25.34	0

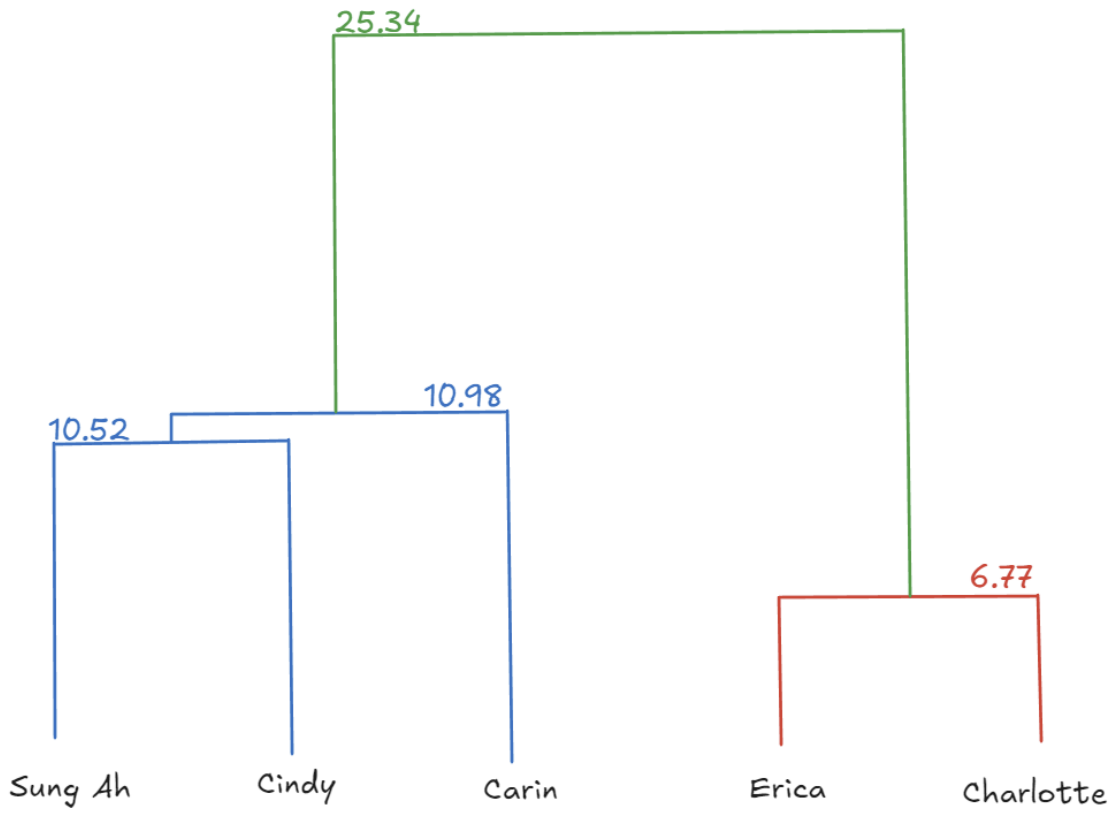
Next iteration:

Golfer	Sung Ah, Cindy	Carin	Erica, Charlotte
Sung Ah, Cindy	0	**10.98**	25.34
Carin	**10.98**	0	15.47
Erica, Charlotte	25.34	15.47	0

This gives us the last iteration:

Golfer	Sung Ah, Cindy, Carin	Erica, Charlotte
Sung Ah, Cindy, Carin	0	25.34
Erica, Charlotte	25.34	0

From this, we can construct the dendrogram:



Cutting off at just above a height of **10.98** would result in 2 clusters.

Part c

The Rand index for the clustering of Part a, A, and Part b, B, can be found, given that there are 5 choose 2 pairs (10):

$$\text{Rand}(A, B) = \frac{10}{10} = 1.0$$

We know this is 1 because both the K-means and Agglomerative clustering placed the golfers in the same groups: Erica and Charlotte in 1 group and Sung Ah, Cindy, and Carin in the other → so, every pair of golfers is in agreement → a Rand index of 1 ⇒ identical clustering results.

Question 3 - Standardization

Part a

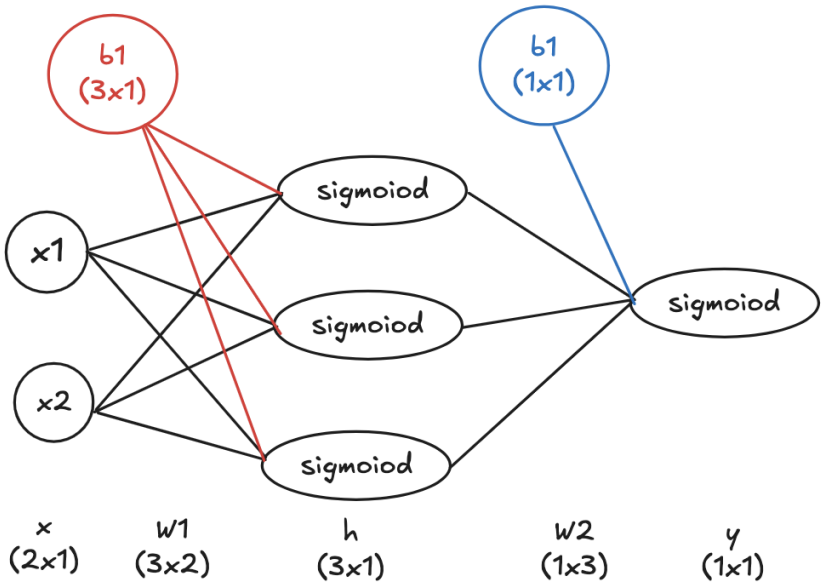
For this scenario, it makes sense to mean-center and scale the data because it is likely that there are many different types of houses with many features/variables being incredibly large or distributed unevenly. Additionally, the large number of features implies that PCA would likely work well here to reduce dimensionality by projecting the highest variability onto a handful of orthogonal axes instead of the possibly hundreds of features. Given this, it makes sense to standardize the data as PCA can be distribution sensitive when calculating variance due to extreme bias, outliers, or other phenomena in the housing data. However, clustering would not be a good approach here as the goal is to predict housing price instead of classifying - though you could use K-means regression while maintaining cluster centroids.

Part b

In this scenario, we likely do NOT want to mean-center as the data is heavily dependent on the distribution of survey responses so we should maintain that distribution as closely to the original as possible. As for scaling, 0-5 is a fairly small range of values for computation so it wouldn't make sense to scale down the entire dataset for clustering, but if we were to do something like classification via NNs, I would probably suggest scaling down to a range of 0-1 for faster floating point computations. PCA may also be beneficial here due to the "dozens of questions" in the survey suggesting possibly many features/columns/variables. Given this, we likely want to reduce dimensionality by projecting the features with the highest variance along a few PCs. Clustering here is also a good idea as we don't really have labels for supervised classification, so an unsupervised clustering algorithm, like agglomerative or K-means clustering would work well to elicit demographic categories from the survey responses.

Question 4 - Backprop

Part a



Part b

The expanded formula for the NN outputs is:

$$\hat{y} = \sigma_2\Big(W_2 \cdot \sigma_1\big(W_1\vec{x} + \vec{b_1}\big) + \vec{b_2}\Big)$$

We can abstract this with the following few equations:

$$z_1 = W_1\vec{x} + \vec{b_1} \quad \text{and} \quad \vec{h_1} = \sigma_1(z_1) \quad \text{and} \quad z_2 = W_2\vec{h_1} + \vec{b_2}$$

s.t.

$$\hat{y} = \sigma_2(z_2)$$

Part c

There may be rounding errors due to low float precision.

i

Given the values for the weights, biases, inputs, and true observation, we can conduct the forward pass via the closed form:

$$\hat{y} = \sigma_2\left(W_2 \cdot \sigma_1\left(\begin{bmatrix} 0.13315865 & 0.0715279 \\ -0.15454003 & -0.00083838 \\ 0.0621336 & -0.07200856 \end{bmatrix}\begin{bmatrix} -0.01746002 \\ 0.04330262 \end{bmatrix}\right)\right)$$

$$= \sigma_2\left(W_2 \cdot \sigma_1\left(\begin{bmatrix} 0.0007723928 \\ 0.0026619679 \\ -0.0042030131 \end{bmatrix}\right)\right)$$

$$= \sigma_2\left(\begin{bmatrix} 0.02655116 & 0.01085485 & 0.00042914 \end{bmatrix}\begin{bmatrix} 0.5001931190 \\ 0.5006654859 \\ 0.4989492297 \end{bmatrix}\right)$$

$$= \sigma_2\left(\left[0.0189294741\right]\right) = 0.5047321916$$

ii

We can get the formula for the gradients of the loss wrt. each parameter via element-wise partials (because the gradient of a parameter with value 0 is also 0, we can omit calculating the gradients wrt biases):

The gradient of the loss wrt. any element of matrix W_2 can be found with the following formula where $* \in [1, 3]$ represents an arbitrary column of the matrix

$$\nabla_{W_2^{(1,*)}} L = \frac{\partial L}{\partial W_2^{(1,*)}} = \frac{\partial L}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial \sigma_2^{(1,1)}} \frac{\partial \sigma_2^{(1,1)}}{\partial W_2^{(1,*)}}$$

The gradient of the loss wrt. any element of matrix W_1 can be found with the following formula where $d \in [1, 3], n \in [1, 2]$ represents an arbitrary row and column of the matrix:

$$\nabla_{W_1^{(d,n)}} L = \frac{\partial L}{\partial W_1^{(d,n)}} = \frac{\partial L}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial \sigma_2^{(1,1)}} \frac{\partial \sigma_2^{(1,1)}}{\partial \sigma_1^{(d,1)}} \frac{\partial \sigma_1^{(d,1)}}{\partial W_1^{(d,n)}}$$

The gradient of the loss wrt. any element of the inputs $\vec{x}$ can be found by the following formula where $n \in [1, 2]$ represents an

arbitrary row of the input vector:

$$\nabla_{x^{(n,1)}} L = \frac{\partial L}{\partial x^{(n,1)}} = \sum_{d=1}^{3} \frac{\partial L}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial \sigma_2^{(1,1)}} \frac{\partial \sigma_2^{(1,1)}}{\partial \sigma_1^{(d,1)}} \frac{\partial \sigma_1^{(d,1)}}{\partial x^{(n,1)}}$$

Now, we can find what each factor in the gradients above evaluate to (abstractly):

$$\frac{\partial L}{\partial \hat{y}} = 2(\hat{y} - y)$$

$$\frac{\partial \hat{y}}{\partial \sigma_2} = 1$$

$$\frac{\partial \sigma_2}{\partial \sigma_1} = \frac{\partial \sigma_2}{\partial z_2} \frac{\partial z_2}{\partial h_1} \frac{\partial h_1}{\partial \sigma_1} = W_2 \cdot \sigma_2(z_2) \cdot (1 - \sigma_2(z_2))$$

$$\frac{\partial \sigma_2}{\partial W_2} = \frac{\partial \sigma_2}{\partial z_2} \frac{\partial z_2}{\partial W_2} = h_1 \cdot \sigma_2(z_2) \cdot (1 - \sigma_2(z_2))$$

$$\frac{\partial \sigma_1}{\partial W_1} = \frac{\partial \sigma_1}{\partial z_1} \frac{\partial z_1}{\partial W_1} = x \cdot \sigma_1(z_1) \cdot \big(1 - \sigma_1(z_1)\big)$$

$$\frac{\partial \sigma_1}{\partial x} = \frac{\partial \sigma_1}{\partial z_1} \frac{\partial z_1}{\partial x} = W_1 \cdot \sigma_1(z_1) \cdot \big(1 - \sigma_1(z_1)\big)$$

Now, using these "net input"-abstracted partials, we get the following gradients:

$$\nabla_{W_2^{(1,*)}} L = \Big[2(\hat{y} - y) \cdot h_1 \cdot \sigma_2(z_2) \cdot (1 - \sigma_2(z_2)) \Big]^{(1,*)}$$

$$\nabla_{W_1^{(d,n)}} L = \Big[2(\hat{y} - y) \cdot W_2 \cdot \sigma_2(z_2) \cdot (1 - \sigma_2(z_2)) \cdot x \cdot \sigma_1(z_1) \cdot \big(1 - \sigma_1(z_1)\big) \Big]^{(d,n)}$$

$$\nabla_{x^{(n,1)}} L = \sum_{d=1}^{3} \Big[2(\hat{y} - y) \cdot W_2 \cdot \sigma_2(z_2) \cdot (1 - \sigma_2(z_2)) \cdot W_1 \cdot \sigma_1(z_1) \cdot \big(1 - \sigma_1(z_1)\big) \Big]^{(d,n)}$$

$$\nabla_{b_1^{(*,1)}} L = \nabla_{b_2^{(1,1)}} L = 0$$

iii

Now, we can weight update $W_1^{(1,1)}$ using the gradients we found above and a learning rate of $\eta = 0.1$:

$$W_1^{(1,1)} \leftarrow W_1^{(1,1)} - \eta \nabla_{W_1^{(1,1)}} L$$

Where the gradient is given by:

$$\nabla_{W_1^{(1,1)}} L = \left[2(\hat{y} - y) \cdot W_2 \cdot \sigma_2(z_2) \cdot (1 - \sigma_2(z_2)) \cdot x \cdot \sigma_1(z_1) \cdot \big(1 - \sigma_1(z_1)\big) \right]^{(1,1)}$$

Given the sample:

$$x = [-0.01746002 \quad 0.04330262] \quad \text{and} \quad y = [1]$$

and weights + the forward pass prediction we found in part i:

$$W_{1,\text{old}}^{(1,1)} = 0.13315865$$

$$\hat{y} = 0.5000118998765415$$

We can find the gradient using the formula in part ii:

$$\nabla_{W_1^{(1,1)}} L = 2.8697189919 \times 10^{-5}$$

Now, the updated weight is:

$$W_{1,\text{new}}^{(1,1)} = 0.13315865 - 0.1 \times 2.8697189919 \times 10^{-5} \approx 0.1331557781$$