CSC 390: Topics in Artificial Intelligence

Midterm: Fall 2016

Due: Wednesday, October 26 at 5pm

- This is a take-home exam with unlimited time from when it is out to when it is due.
- It is open-notes, so you may use any course materials. If you use any online resources that haven't been part of this class, please cite them.
- No communication about the exam with anyone in the class (or outside the class).
- If there is a clarification I think should be made to the entire class, I'll post it on Piazza.
- I will still have office hours Thursday 4-5pm and Monday 5-6pm, but I might not say much!
- If you are not in class to pick up an exam due to Grace Hopper, I will have copies outside my office. You are also welcome to print the exam on your own, it will be available on the course website.
- Turn in your exam to me in person or put it under the door of my office if I am not there (not in the bin outside).
- If you are unable to make it to my office to turn in the exam before the deadline, email me to make other arrangements such as scanning your exam.
- If you are unable to make progress on any part of the exam, tell me what you tried; describe your thought process.

Name	Solution Set (sketches)

Part 1	/25	✓
Part 2	/15	✓
Part 3	/25	✓
Part 4	/20	✓
Part 5	/15	✓
Total	/100	

Part 1: Vocab and Written Communication

(a) Is the nearest-neighbors method an example of supervised or unsupervised learning? Justify your answer.

NN is an example of supervised learning because we use the true labels of training data to classify new (test) data.

(b) Consider the data analysis we did in Homework 4. Was this an example of supervised or unsupervised learning? Justify your answer.

Homework 4 was an example of unsupervised learning because we never used the labels of the data to train / fit our models (PCA, k-means).

(c) Is UPGMA an example of agglomerative or divisive clustering? Justify your answer.

UPGMA is an example of agglomerative clustering because we start with each data point in its own cluster and gradually build the hierarchy by merging clusters.

(d) How can labeled *test* data be used to assess the accuracy of k-means? Does this qualify as supervised learning? Would the training data need to be labeled to use this approach? Justify your answer.

To classify test data after k-means has been run on training data, we can compute the distance between each new test data point and each of the k cluster means. Then assign it the label of the closest mean. If these test data points have true labels, we can compare these to the assigned labels to assess the accuracy. This is not supervised learning since labels were not used to fit the model. Training data does not need to be labeled but it can be.

(e) Consider the training dataset:

$$X = \begin{bmatrix} 2 & -1 & 6 \\ 0 & 3 & -3 \end{bmatrix} \begin{matrix} \leftarrow x_1 \\ \leftarrow x_2 \end{matrix}$$

with labels $y = [0, 1]$. Given a new test data point $x_{\text{test}} = [-1, 1, 2]$, what label would you assign it using a 1-nearest neighbors approach with Euclidean distance? Show your work to justify your answer.

$$d(x_1, x_{test}) = \sqrt{(2--1)^2 + (-1-1)^2 + (6-2)^2}$$
$$= \sqrt{9+4+16} = \boxed{\sqrt{29}}$$

$$d(x_2, x_{test}) = \sqrt{(0--1)^2 + (3-1)^2 + (-3-2)^2}$$
$$= \sqrt{1+4+25} = \boxed{\sqrt{30}}$$

The closest neighbor to x_{test} is x_1, which is labeled 0.

$\Rightarrow$ Assign x_{test} the $\boxed{\text{label } 0}$ as well.

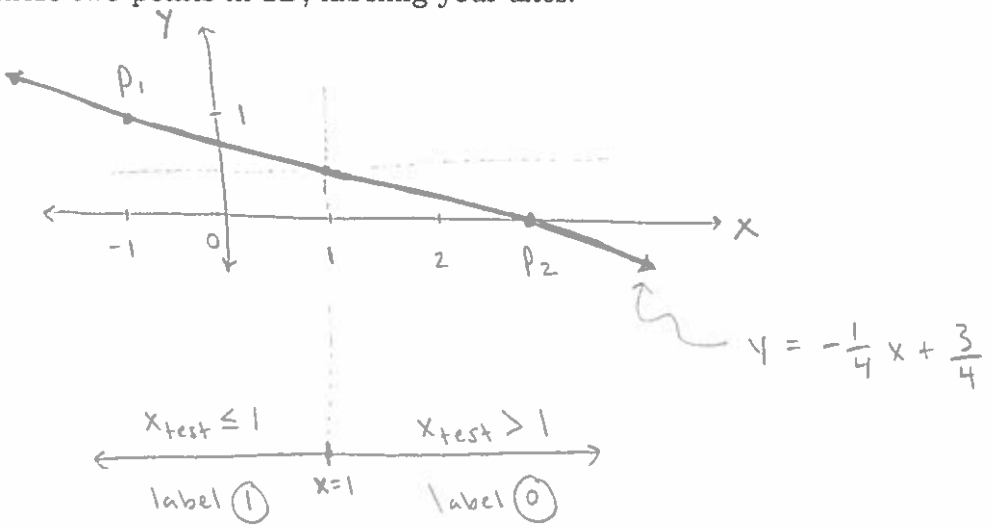
(f) In k-nearest neighbors we have one parameter, k. In linear regression, we have $p+1$ parameters, where p is the number of features ($+1$ for the intercept). In these two methods, which parameters are chosen in advance by the user? What data is used to find the parameters *not* chosen by the user? Explain your answer, using the concepts of "fitting a model" and "training data".

In k-NN, k must be chosen ahead of time by the user. In linear regression, the p+1 parameters are determined by both the features and the labels of the training data. This process of finding these p+1 parameters is called "fitting the [linear (in this case)] model." In nearest neighbors, there isn't a parametric model to fit, the only parameter is chosen in advance, before seeing the data.

Part 2: Linear Regression

Consider the labeled training dataset $p_1 = (x_1, y_1) = (-1, 1)$ and $p_2 = (x_2, y_2) = (3, 0)$, where each point has one feature and one class label in $\{0, 1\}$.

(a) Plot these two points in 2D, labeling your axes.



(b) After running linear regression on this dataset, what is the resulting linear equation? Show this line on your graph above. How many parameters does this linear model have?

slope: $m = \frac{0-1}{3--1} = \frac{-1}{4}$

point slope form:

$y = -\frac{1}{4}(x-3)$

$\boxed{y = -\frac{1}{4}x + \frac{3}{4}}$

This linear model has **2 parameters** (slope & intercept).

(c) Given a new test data point x_{test} (again with one feature), what rule does this model give us about how x_{test} should be classified?

Plug in $y = \frac{1}{2}$: $\frac{1}{2} = -\frac{1}{4}x + \frac{3}{4}$

solve for x: $-\frac{1}{4} = -\frac{1}{4}x \Rightarrow \boxed{x = 1}$

$\Rightarrow$ classification rule:

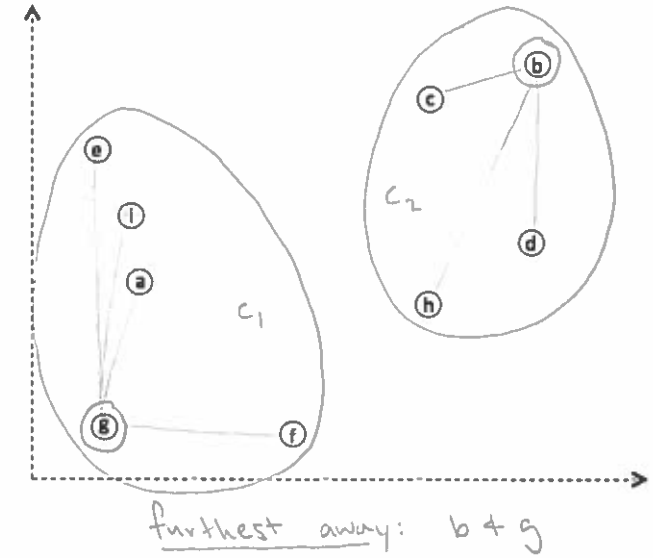
$\boxed{\begin{array}{ll} x_{test} \leq 1 & \Rightarrow \text{label } 1 \\ x_{test} > 1 & \Rightarrow \text{label } 0 \end{array}}$

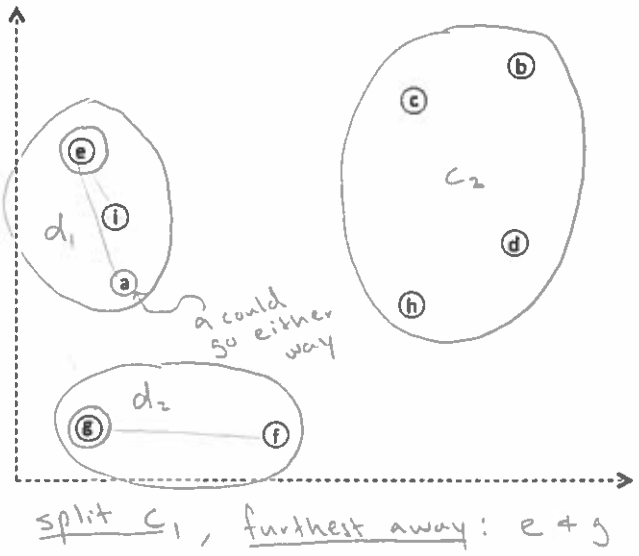
Part 3: Hierarchical Clustering

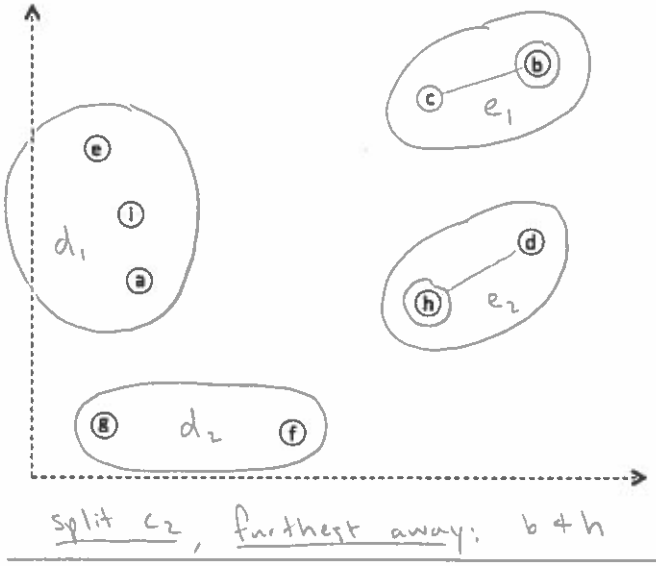
Inspired by UPGMA, you decide to create a clustering algorithm that is top-down instead of bottom-up. So instead of first choosing the two points that are the closest to each other, in this new algorithm we will choose the two points that are *furthest away* from each other. This algorithm will proceed as follows. The data points will begin in one cluster. Then iterate the following steps until all points are in their own clusters:

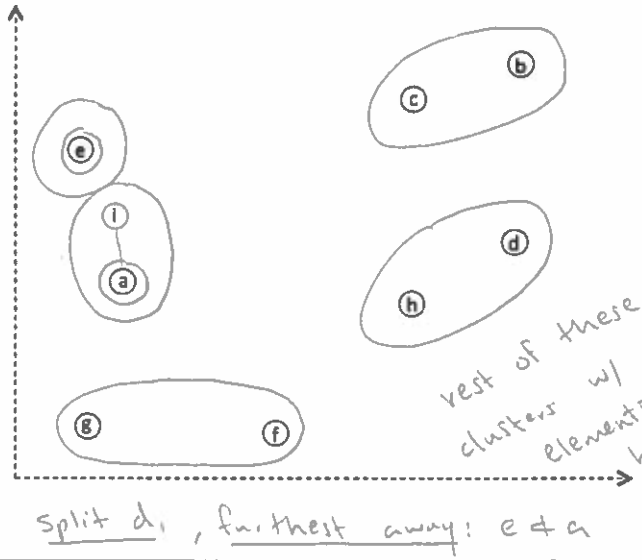
- Select a cluster c_j to split. Find the two points that are the furthest from each other within this cluster. Denote these as x_{m1} and x_{m2}. These two points will now be in different clusters.
- For each point within c_j, compare its distance from x_{m1} and x_{m2} and assign it to the closest point. In this way we break up the cluster c_j into two new clusters: c_{j1} and c_{j2}.

(a) Given the example dataset in 2D below, run this clustering algorithm. The dataset is replicated a few times to show different iterations. To decide which cluster to split, use the largest current cluster.

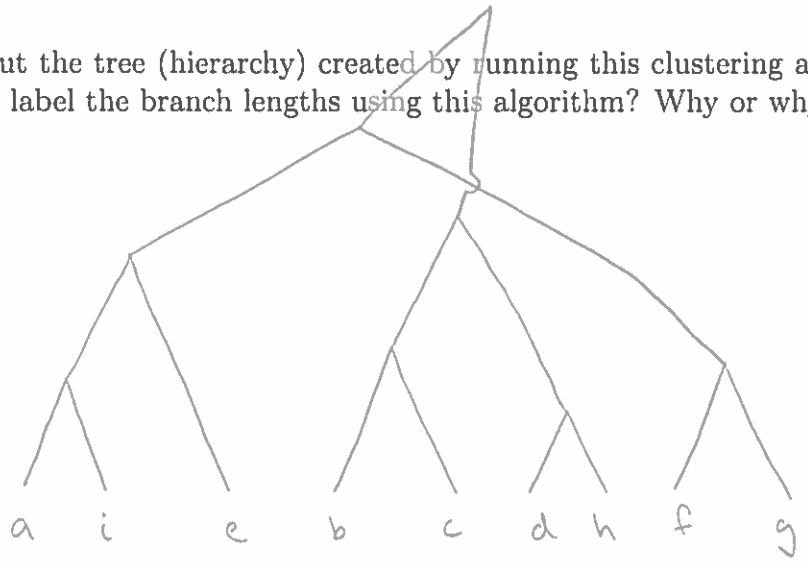








(b) Draw out the tree (hierarchy) created by running this clustering algorithm on these points. Can we label the branch lengths using this algorithm? Why or why not?

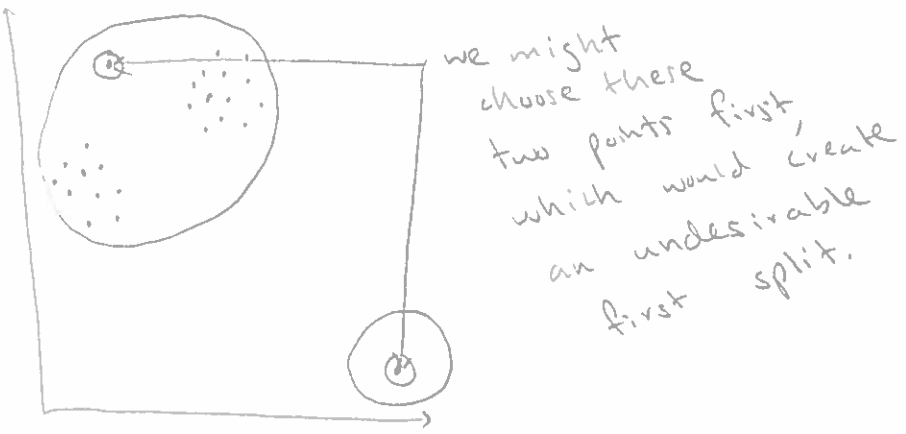


As the algorithm is described, we don't have a general notion of distances between clusters, so we can't label the branch lengths. However, we could take the distance between clusters to be the average of the distances between all pairs, which could start to give us an idea of the branch lengths, but it could create inconsistencies, unlike UPGMA.

(c) What could go wrong when using this algorithm? Sketch an example dataset where this algorithm would not achieve a desirable clustering.

This algorithm could be very sensitive to outliers.

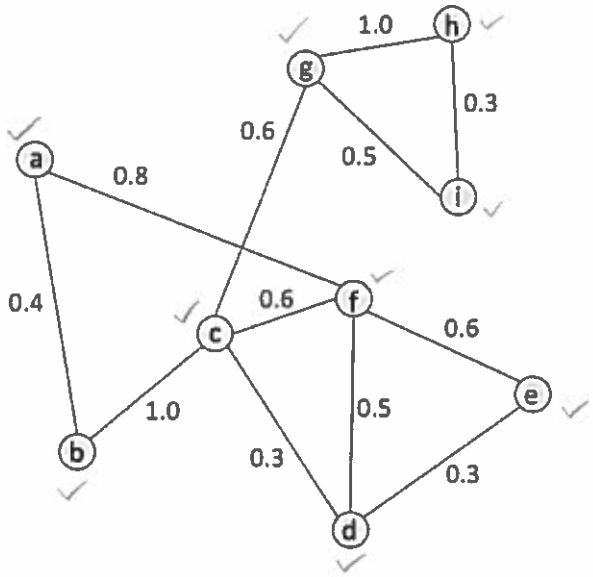
Example:

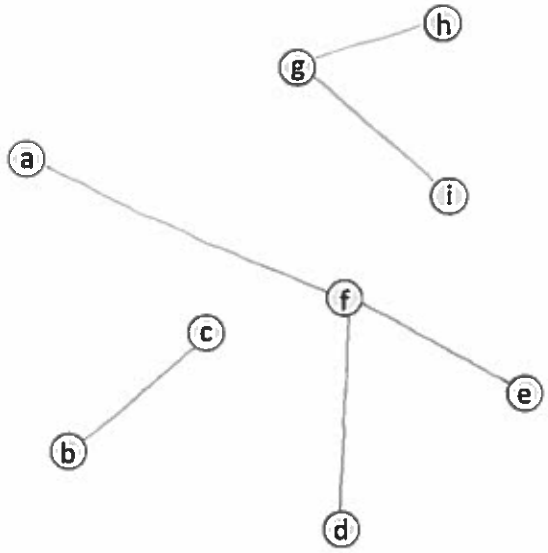


Part 4: Clustering Literature

The figure below shows the β-similarity graph for $\beta = 0.3$, for a set of data points $X = [a, b, c, d, e, f, g, h, i]$. Each value is a measure of similarity between two points (so higher values denote more similar points). Edges are drawn only if the similarity value is at least β.

(a) Using the β-similarity graph on the left, create the max-S graph on the right, making sure to consider each point in turn.





(b) Using the *dynamic hierarchical compact* algorithm framework from Paper 2 (i.e. no overlapping clusters), how many clusters (k) should we obtain from the max-S graph? Write out each of these clusters with the points assigned to them. Denote this clustering $\mathcal{C} = \{c_1, c_2, \cdots, c_k\}$.

We obtain $k = 3$ clusters:

$c_1 = \{b, c\}$

$c_2 = \{a, f, d, e\}$

$c_3 = \{g, h, i\}$

(c) A different clustering algorithm obtains the following clustering $\mathcal{D} = \{d_1, d_2\}$, where

$$d_1 = \{a, d, e, f\} \quad \text{and} \quad d_2 = \{b, c, g, h, i\}.$$

We will compare these two clusterings using the *Rand index*. First, compute A, the number of *pairs* of points that are assigned the *same* cluster in both $\mathcal{C}$ and $\mathcal{D}$.

note: $c_2 = d_1$, $d_2 = c_1 \cup c_3$

$\mathcal{C}$: $c_1 = \{b, c\}$, $c_2 = \{a, f, d, e\}$, $c_3 = \{g, h, i\}$

$$A = \underbrace{\binom{4}{2}}_{c_2} + \underbrace{\binom{3}{2}}_{c_3} + \underbrace{\binom{2}{2}}_{c_1} = 6 + 3 + 1 = 10$$

$$\Rightarrow \boxed{A = 10}$$

(d) Next compute B, the number of pairs of points that are assigned to *different* clusters in *both* $\mathcal{C}$ and $\mathcal{D}$.

All pairs where one element is in d_1 & one is in d_2 are in different clusters in both $\mathcal{C}$ & $\mathcal{D}$:

$$\boxed{B = 4 \cdot 5 = 20}$$

(e) Using these results, compute the Rand index:

$$R = \frac{A + B}{\binom{n}{2}}$$

where n is the number of points ($n = 9$ here). Interpret this quantitative result.

$$R = \frac{10 + 20}{\frac{9 \cdot 8}{2}} = \frac{30}{36} = \boxed{\frac{5}{6}}$$

Since R is quite close to 1, these clusterings are quite similar.

Part 5: Dimensionality Reduction

Consider the following dataset with $m = 3$ data points and $p = 4$ features:

$$X = \begin{array}{c} \begin{matrix} f_1 & f_2 & f_3 & f_4 \end{matrix} \\ \begin{bmatrix} -3 & 0 & 2 & 1 \\ 1 & -1 & 0 & 2 \\ 2 & 1 & -2 & -3 \end{bmatrix} \end{array}$$

Note that the features have been normalized (i.e. each column has mean 0). Without feature selection / dimensionality reduction, we cannot visualize this dataset.

(a) Below is a portion of the covariance matrix A for this dataset. Fill in the missing entries, showing your work. Will the diagonal always contain non-negative values?

* $\text{cov}(g, f_i) = \frac{1}{n-1} \sum_{i=1}^{n} (f_i - \bar{f})(g_i - \bar{g})$
* $\bar{f} = \bar{g} = 0$ for all features

$$A = \begin{bmatrix} 7 & \frac{1}{2} & -5 & -\frac{7}{2} \\ \frac{1}{2} & 1 & -1 & -\frac{5}{2} \\ -5 & -1 & 4 & 4 \\ -\frac{7}{2} & -\frac{5}{2} & 4 & 7 \end{bmatrix}$$

symmetric

f_1 f_2 f_3 f_4 } four features

- $\text{cov}(f_2, f_3) = \frac{1}{2}(0 \cdot 2 - 1 \cdot 0 + 1 \cdot (-2)) = \frac{1}{2}(-2) = -1$
- $\text{cov}(f_2, f_4) = \frac{1}{2}(0 \cdot 1 - 1 \cdot 2 + 1 \cdot (-3)) = -\frac{5}{2}$
- $\text{var}(f_1) = \frac{1}{2}((-3)^2 + 1^2 + 2^2) = \frac{1}{2}(9 + 1 + 4) = 7$

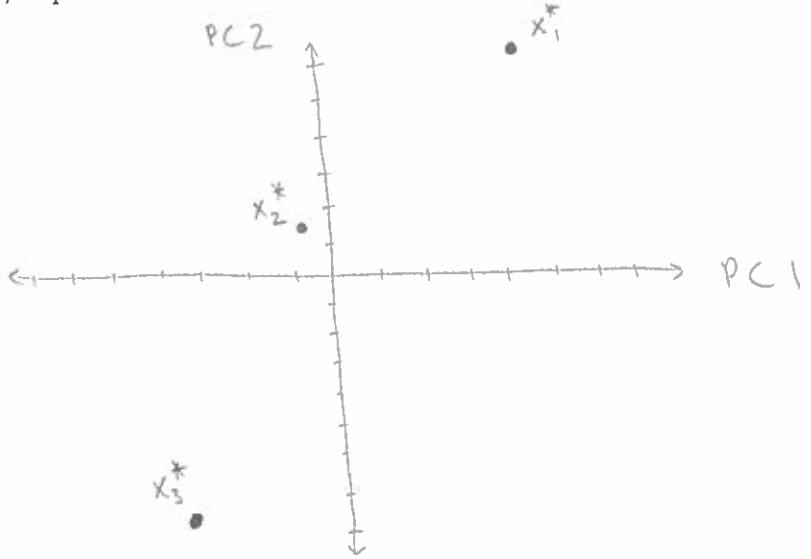
(b) The eigenvalues of A are: $\lambda_1 = \frac{1}{2}(19 + 4\sqrt{7})$, $\lambda_2 = \frac{1}{2}(19 - 4\sqrt{7})$, $\lambda_3 = 0$, $\lambda_4 = 0$. Which eigenvalue corresponds to the first principal component (PC)? Which eigenvalue corresponds to the second PC? What do the zero eigenvalues tell us about this dataset?

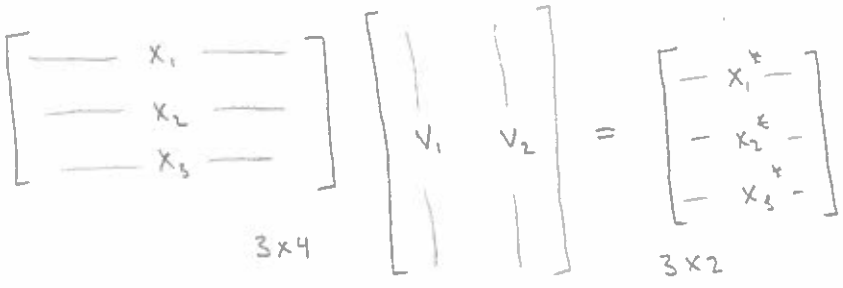
- λ_1 corresponds to the first PC since it is the largest.
- λ_2 corresponds to the second PC since it is second largest.
- The 2 zero eigenvalues tell us that all the variation in this data set can be captured in 2 dimensions, we are not losing anything by reducing to 2D.

(c) The resulting transformed dataset is approximately:

$$X^* \approx \begin{bmatrix} 4.4 & 6.2 \\ -0.6 & 1.6 \\ -3.8 & -7.8 \end{bmatrix} \begin{matrix} x_1^* \\ x_2^* \\ x_3^* \end{matrix}$$

Plot these points on a graph (approximately), labeling your axes. Then write out the matrix multiplication problem that results in X^* (you don't need to use any numerical values, just show the matrices involved and how the dimensions work out). Using this matrix multiplication set up, explain how each of these two new features is a linear combination of the original features.





matrix of the first 2 eigenvectors, denote W_2

original data

$\Rightarrow$ $X W_2 = X^*$ ← transformed data