Unsupervised Learning Problems

1 Unsupervised Learning Problem

1.1 Visualization

Alice and Bob, your colleagues from the astrophysics department, have given you a collection of astronomical data[1] describing exoplanets in different star systems. Each exoplanet is described by the distance from its orbiting star in AU (astronomical units), its mass (as multiples of the Earth), and the degree of light reflection (as an albedo integer). See table below.

	AU from star	Mass	Albedo
HD 209458 b	2	3	7
HD 189733 b	5	3	3
51 Pegasi b	7	2	5
PSR B1257+12 B	3	5	6
PSR B1257+12 C	5	4	5
OGLE-TR-56 b	7	4	3
Fomalhaut b	3	3	8
2M1207 b	4	3	7

Some of these data (about half) have been collected using the *transit detection method* and others (about half) using an *infrared detection methods*. Alice and Bob know that these two methods are sensitive to exoplanets with different features, but they do not know which sample has been collected with which method.

Alice argues that looking at *AU from the star* and *albedo* may help them infer which observations were performed with which techniques; Bob holds that looking at *AU from the star* and *mass* may provide a better perspective to group the exoplanets by their discovery method.

Plot the data first according to Alice's hypothesis and then Bob's hypothesis. Which hypothesis seems more likely?

1.2 K-Means

To give more grounding to your conclusions, you decide to run the *k-means algorithm* on your data using two clusters, one for each detection method. Let us call one cluster the blue cluster, and the other one the red cluster.

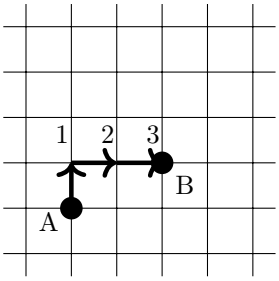
Run three iterations of k-means (assignment, recomputation of the centroids) on Alice's and Bob's data.

Initialize the blue cluster at (3,2), and the red cluster at (8,4).

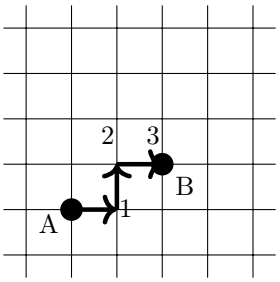
In the assignment phase, use as a *distance function* the *Manhattan distance* $D_{Man}[x_i, x_j]$, that gives you the number of straight segments necessary to get from one point to the other, for example:

If a point is the same distance from the center of both clusters, assign it to the blue cluster.

[1] Exoplanet names are real. All the other details are made up.



(a) Manhattan distance



(b) Notice that the distance is independent from the path.

In the recomputation of the centroids, round the values of the nearest integer:

$$5.3 \to 5$$
$$2.5 \to 3$$
$$3.8 \to 4$$

How do your result with your conclusions from the visualization exercise?

1.3 Quantitative Evaluation

Bob and Alice look very interested in your results: it seems that clustering based on a given pair of features is better than clustering on another set of features. However, they are uneasy accepting a solution based on an intuitive visualization. They ask if your results may be given a quantitative explanation.

You think that an easy way would be to compute the *separation* between the clusters, that is computing the distance between the blue point that is closest to the red cluster and the red point that is closer to the blue cluster. This measure would quantify the gap between the two cluster.

Compute the separation for the clustering of Alice's data and Bob's data.

This measure would quantifies the gap between the two cluster.

What would you conclude from the computation of cluster separation?

Yet, you feel this measure is not very robust.

What problem could you imagine having when using cluster separation?

You ask around your colleagues, and Yoshua explains to you that there are two important measure to evaluate clustering: the *inter-cluster distance*, measuring how separate two clusters are, and the *intra-cluster distance*, measuring how compact a cluster is.

Compute the inter-cluster distance for the clustering of Alice's data and Bob's data (do not round to integers).

To compute inter-cluster distance simply compute the distance between the centroid of the red and blue cluster: $D_{inter}\left[c_{blue}, c_{red}\right] = D_{Man}\left[t_{blue}, t_{red}\right]$, where c is a cluster and t is a centroid.

Compute the intra-cluster distance for the clustering of Alice's data and Bob's data (do not round to integers).

Differently from the inter-cluster distance, the intra-cluster must be computed for each cluster individually. For each cluster, red or blue, compute the average distance of all the cluster points

from the cluster center. For the blue cluster: $D_{intra}\left[c_{blue}\right] = \frac{1}{N_{blue}} \sum_{x \in c_{blue}} D\left[x, t_{blue}\right]$; similarly for the red cluster. Average then the intra-cluster distance of the blue and red cluster to get the overall intra-cluster distance for Alice and Bob: $\frac{1}{2}\left(D_{intra}\left[c_{blue}\right] + D_{intra}\left[c_{red}\right]\right)$.

A good cluster is a cluster that clumps its point tightly close to each other, and that is far removed from other cluster. It is natural to assess the goodness of your clustering as the ratio between inter-cluster distance (which you want to be big) and intra-cluster distance (which you want to be small).

Compute the ratio of inter-cluster distance and intra-cluster distance for the clustering of Alice's data and Bob's data (do not round to integers). How does this confirm/reject your previous conclusions?

1.4 Processing new data

Alice and Bob are happy with your solution, and decide to adopt the clustering you argued being the best one. From now on, we will use only the clustering that you proved being the best. Now new data has come in:

	AU from star	Mass	Albedo
Beta Pictoris c	9	3	6
K2-282c	6	5	7
Kepler-1658b	2	2	8

Start from the chosen clustering, plot the new data points and assign them to the correct cluster.

1.5 Outliers

There is a further recordings, coming from another institution, that Alice and Bob would like to process:

	AU from star	Mass	Albedo
Luyten 98-59 d	22	3	3

Alice and Bob are not certain about the quality of this recording and ask your opinion.

Use the chosen clustering, plot the new data point. What do you think about this observation?

For your own interest, you decide to analyze how this new data point will affect the clustering process.

Restart from the original data set of eight data points; place the centroids of the two clusters as you computed them at then end of Section 1.2; add the new data on Luyten 98-59 d and run two iterations of the k-mean algorithm. What happens to the clusters?

1.6 Rescaling

After discussing with other colleagues at a conference, Alice and Bob became suspicious that the recordings of albedo may be wrong. Following the suggestion of Eve, they are thinking about reducing by half all the recorded values of albedo. They present this possible possible change to you, and ask your opinion.

How would you interpret the change that they have proposed?

In particular, they are concerned whether this change would affect your results.

Apply the transformation to the original data. Then run two iterations of k-means with the same initialization used in Section 1.2 on Alice's data. What do you observe?

When halving the observed values of albedo, always round down to the closest integer:

$$3.5 \rightarrow 3$$

Is k-means insensitive to the suggested transformation? If not, how would you tackle this inconsistency?

1.7 PCA

Alice is very happy with the work done so far. However, whenever collecting a new data point, she finds that computing distances from cluster centers in two dimensions is too computationally expensive. She wonders whether you could come up with a single synthetic index to evaluate new data points as they are collected.

You think that a good idea would be to use PCA.

How would you justify the use of PCA with respect to the assumptions of PCA?

After explaining to Alice the reasons to use PCA, you proceed to apply the algorithm to project the data in one dimension.

Plot the original data according to Alice's hypothesis and draw the first eigenvector (to do this, you can just rely on the geometric intuition of eigenvector as the dimension that maximizes the spread of the data in one dimension; you do not need to compute the actual eigenvector and eigenvalue, you can simply draw a graphical approximation on your plot)

Do not round in PCA.

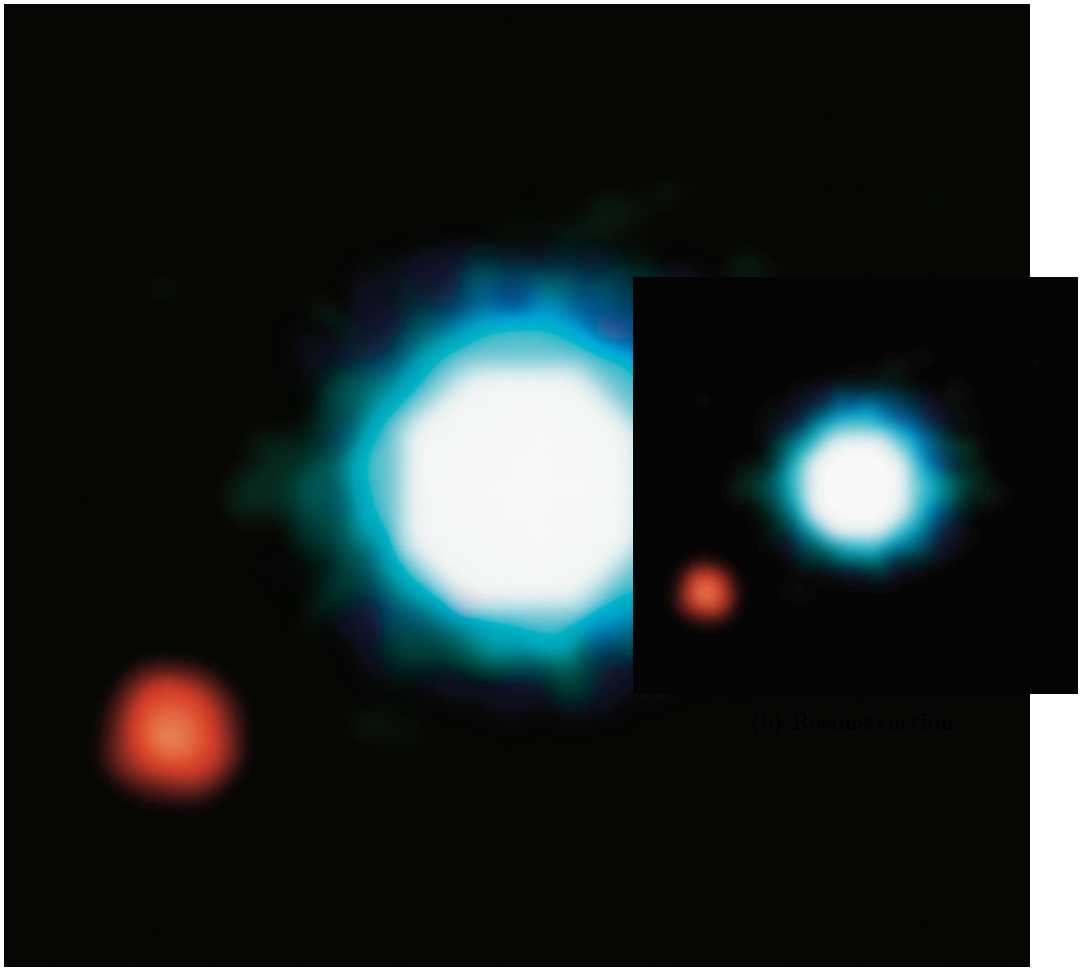
1.8 Autoencoders

Bob has been using standard autoencoders with squared reconstruction loss to compress high-definition images of exoplanets in order to save space. However he has been unhappy with the results, and he has decided to show a sample[2] to you:

Bob complains that the definition of the exoplanet in the middle of the image is very low. He also explains to you that the exoplanet of interest is always the middle of the picture, while other elements around (such as companions or background stars) are of no interest to him. His computational resources are limited, so, ideally he would prefer not to change the architecture of the autoencoder network by adding more layers or more nodes.

How would you recommend changing the autoencoder algorithm to address Bob's challenge?

[2] Figure retrieved at `https://www.eso.org/public/images/26a_big-vlt/`



(a) Original picture.

(b) Reconstruction.