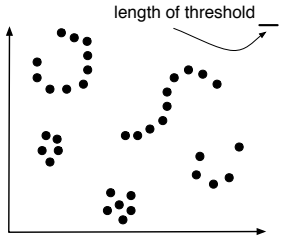
Course: DD2427 - Exercise Set 11

Exercise 1: *Hierarchical clustering*

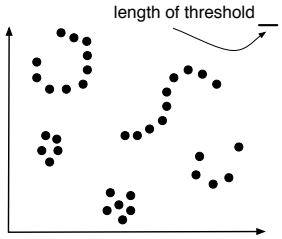
Two popular distance measures between two clusters $\mathcal{D}_i$ and $\mathcal{D}_j$ are

$$d_{\min}(\mathcal{D}_i, \mathcal{D}_j) = \min_{\substack{\mathbf{x}\in\mathcal{D}_i \\ \mathbf{x}'\in\mathcal{D}_j}} \|\mathbf{x}-\mathbf{x}'\| \quad \text{and} \quad d_{\max}(\mathcal{D}_i, \mathcal{D}_j) = \max_{\substack{\mathbf{x}\in\mathcal{D}_i \\ \mathbf{x}'\in\mathcal{D}_j}} \|\mathbf{x}-\mathbf{x}'\|$$

Apply agglomerative clustering with $d_{\min}$ and then $d_{\max}$ used as the measure of distance between clusters to the data-points in the figure below. Terminate the algorithm when the distance between the nearest clusters exceeds the threshold shown. Sketch the clusters at the time of termination.



Clusters using $d_{\min}$



Clusters using $d_{\max}$

For the lecture: 3rd May

Bring your hand written solution to this exercise.

Exercise 2: *k-means clustering (optional)*

In this exercise you will explore k-means clustering to perform an unsupervised clustering task. For this task, you will use the face images you downloaded for Exercise Set 7. You will investigate, admittedly somewhat superficially, two issues

- are eignfaces a good representation for face recognition
- can k-means clustering be used to group pictures of the same person into separate clusters.

So your tasks are as follows:

1. Load the first 50 images, `face00001.bmp`,`face00002.bmp`,..., from the frontal face database. Normalize these images and then use k-means clustering on this data. You can use the *Matlab* function `kmeans`. This called as follows

```
opts = statset('Display', 'final');
[idx, Cs] = kmeans(Xs, k, 'Replicates', 20, 'Options', opts);
```

   where the inputs are - `Xs` an array of size `50`$\times$`(19*19)` (each row corresponds to an image), `k` the number of clusters. The parameter `Replicates` indicates how many runs of k-means clustering is performed each with a different initialization of the cluster centres. The partition with the lowest errror is returned. The outputs are - `idx` a vector of length `50` which contains the cluster number to which example belongs and `Cs` is an array contains the cluster centres.

   Display each cluster when you run this command with `k=10`. I got the clusters that appear in figure 1.

2. Use the code you wrote in `Exercise Set 7` to compute the eigenfaces of the frontal face dataset using all the images except for those that you have just used. When you have obtained the eigenfaces project each of the first `50` images onto the first `nv=30` eigenfaces to obtain an array `Coeffs` of size `50`$\times$`nv`.

3. Repeat the first step, but this time cluster the data using the `Coeffs` data as opposed to the raw pixel data which was used then. Display the clusters found. Using this eigenface representation these are the clusters I found are shown in figure 2.

Note you may or may not get the same clusters as remember the k-means algoritm finds a local minimum solution which is dependent on the starting point and this is chosen at random by *Matlab*. If you have time it may be interesting to see how many different people are contained in the first 50 images and then apply the clustering algorithm with this number of clusters.

Figure 1: **Clusters found using the pixel intensity representation**

Figure 2: **Clusters found using the eigenface representation**