

4. Clustering

4.1. Clustering

4.2. A clustering objective

In Python, we can store the list of vectors in a `numpy` list of N vectors. If we call this list `data`, we can access the i th entry (which is a vector) using `data[i]`. To specify the clusters or group membership, we can use a list of assignments called `grouping`, where `grouping[i]` is the number of group that vector `data[i]` is assigned to. (This is an integer between 1 and k .) (In VMLS, chapter 4, we describe the assignments using a vector c or the subsets G_j .) We can store k cluster representatives as a Python list called `centroids`, with `centroids[j]` the j th cluster representative. (In VMLS we describe the representatives as the vectors z_1, \dots, z_k .)

Group assignment. We define a function to perform group assignment. With given initial value of centroids, we compute the distance between each centroid with each vector and assign the grouping according to the smallest distance. The function then returns a vector of groupings.

```
In [ ]: def group_assignment(data, centroids):
        grouping_vec_c = np.zeros(len(data))
        for i in range(len(data)):
            dist = np.zeros(len(centroids))
            for j in range(len(centroids)):
                dist[j] = np.linalg.norm(data[i] - centroids[j])
            min_dist = min(dist)
            for j in range(len(centroids)):
                if min_dist == dist[j]:
                    grouping_vec_c[i] = j+1
        return grouping_vec_c
```