MapReduce Python

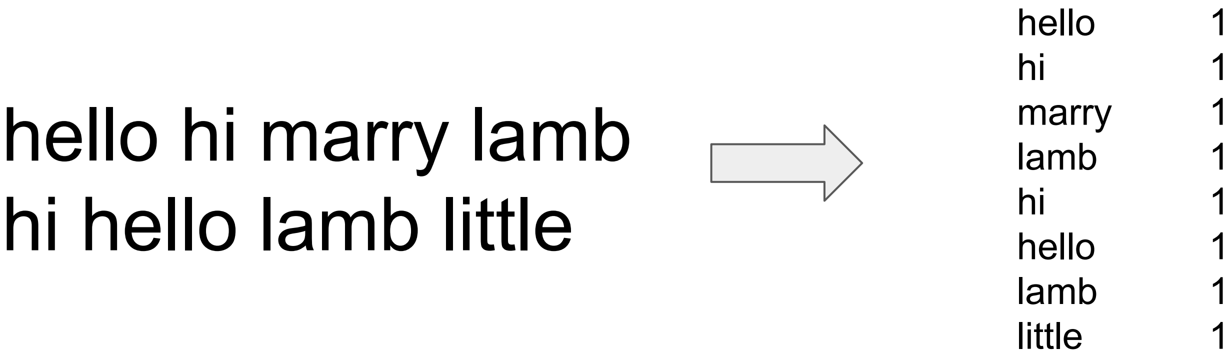
Word Count

Essentially this task can be divided into two parts

MAPPER : It is fairly straight forward the function of the mapper is to iterate over each word and assign a value "1" to it. This is the generation of a key and a value pair which will act as the input to the reducer function.

REDUCER: <key : hello> <value: 1> will be the input for the reducer, the reducer will read the key value pairs and aggregate the number of occurrence for each key, and outputs the results to STDOUT.

Mapper Output



hello hi marry lamb
hi hello lamb little

```
hello       1
hi          1
marry       1
lamb        1
hi          1
hello       1
lamb        1
little      1
```

Reducer Output

hello	1
hi	1
marry	1
lamb	1
hi	1
hello	1
lamb	1
little	1

→

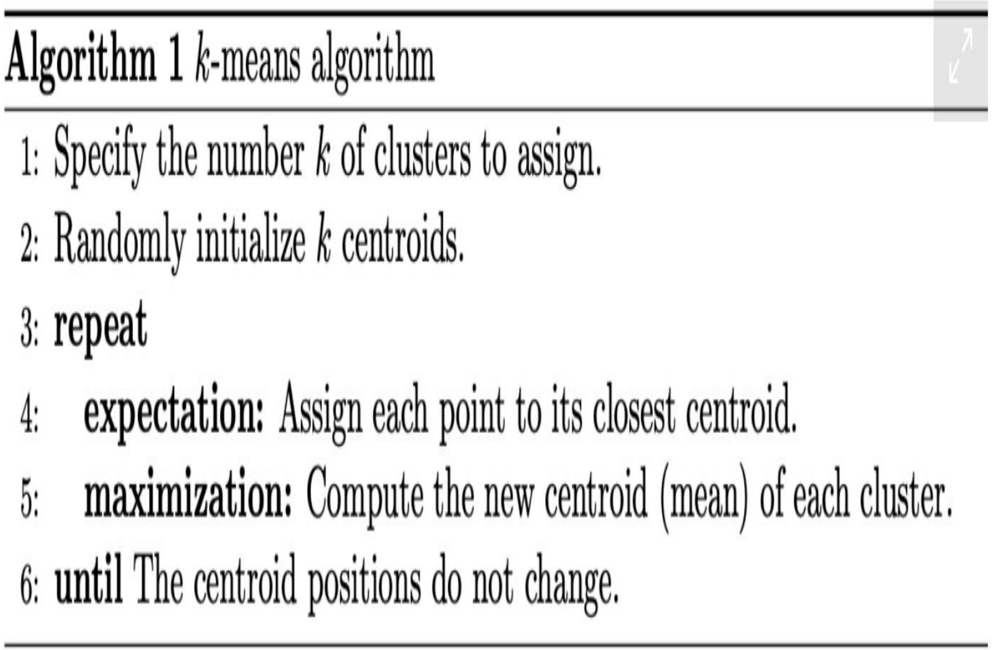
hello	2
hi	2
lamb	2
little	1
marry	1

Hadoop Streaming

- any language written map, reduce program can run on the hadoop cluster
- any map/reduce program as long as it follows from the standard input stdin read, write out to the standard output stdout.
- it is easy to debug on a single machine as it follows a pipeline framework
- provides a rich parameter control for job submission

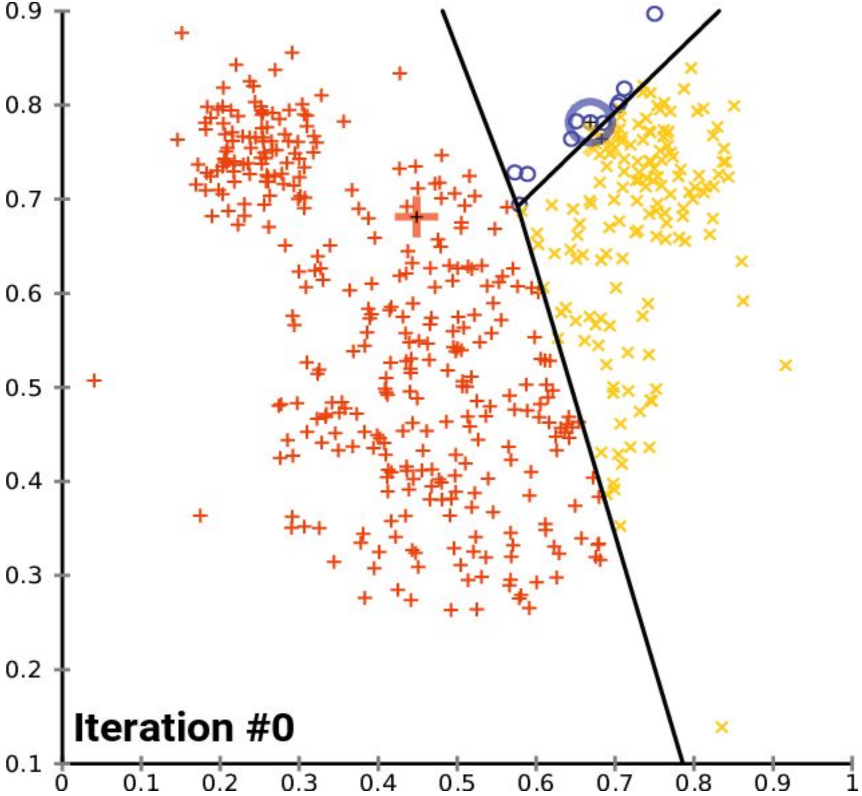
KMeans Clustering

Clustering is a set of techniques used to partition data into groups, or clusters.

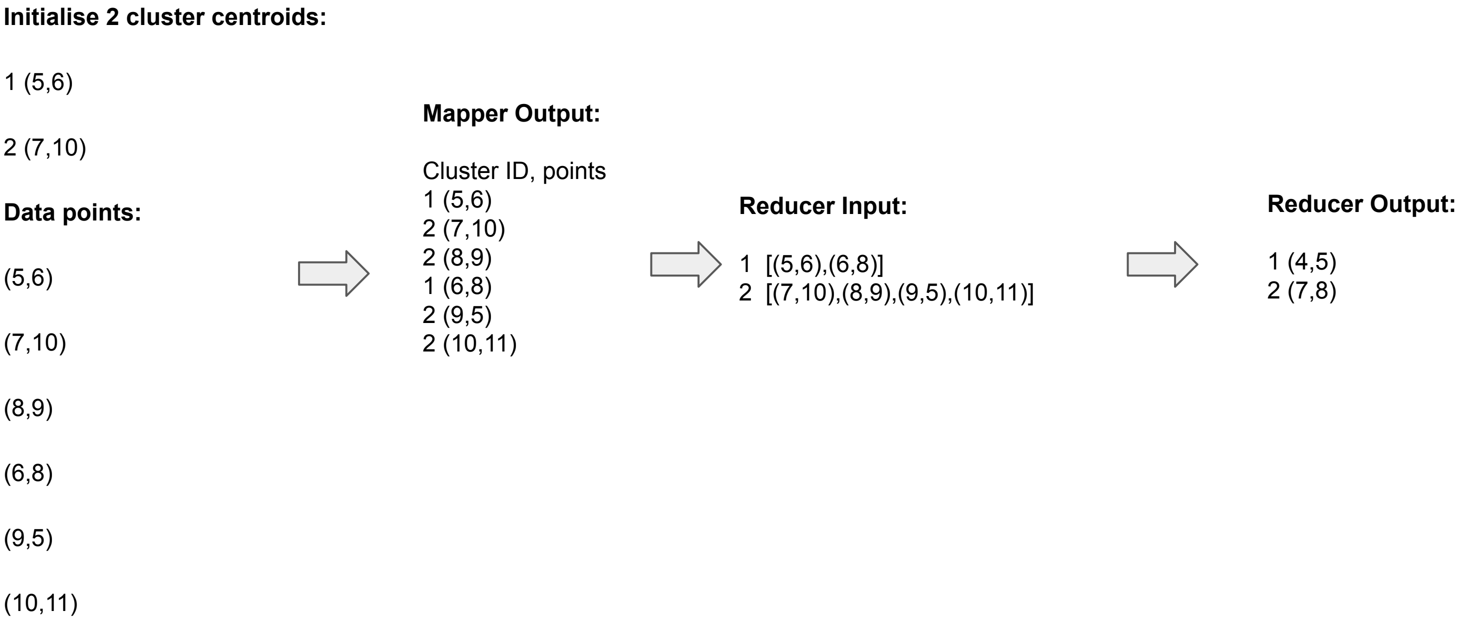


Algorithm 1 k-means algorithm

```
Specify the number $k$ of clusters to assign.
Randomly initialize $k$ centroids.
repeat
    expectation: Assign each point to its closest centroid.
    maximization: Compute the new centroid (mean) of each cluster.
until The centroid positions do not change.
```



Example (1st Iteration)



Initialise 2 cluster centroids:

1 (5,6)

2 (7,10)

Data points:

(5,6)

(7,10)

(8,9)

(6,8)

(9,5)

(10,11)

Mapper Output:

Cluster ID, points
1 (5,6)
2 (7,10)
2 (8,9)
1 (6,8)
2 (9,5)
2 (10,11)

Reducer Input:

1 [(5,6),(6,8)]
2 [(7,10),(8,9),(9,5),(10,11)]

Reducer Output:

1 (4,5)
2 (7,8)