BIRCH

https://www.coursera.org/lecture/cluster-analysis/4-5-birch-a-micro-clustering-based-approach-N06Vq

data stream clustering is defined as the clustering of data that arrive continuously such as telephone records, multimedia data, financial transactions etc. Data stream clustering is usually studied as a streaming algorithm and the objective is, given a sequence of points, to construct a good clustering of the stream, using a small amount of memory and time.

Algorithms

STREAM

BIRCH

COBWEB

C2ICM

k-medoids

CURE

Mini batch kmeans

1. BIRCH – the definition

- **B**alanced
- **I**terative
- **R**educing and
- **C**lustering using
- **H**ierarchies

2 phases

• Phase 1: BIRCH scans the database to build an initial in-memory CF tree, which can be viewed as a multilevel compression of the data that tries to preserve the inherent clustering structure of the data.

• Phase 2: BIRCH applies a (selected) clustering algorithm to cluster the leaf nodes of the CF tree, which removes sparse clusters as outliers and groups dense clusters into larger ones.

Phase 1

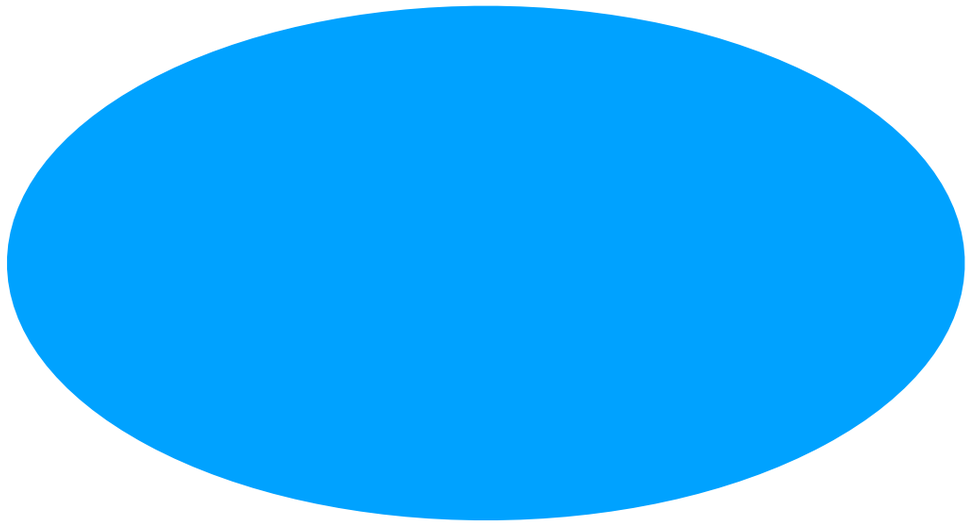
Data set

Row	x
1	0.5
2	0.25
3	0
4	0.65
5	1
6	1.4
7	1.1

Parameters

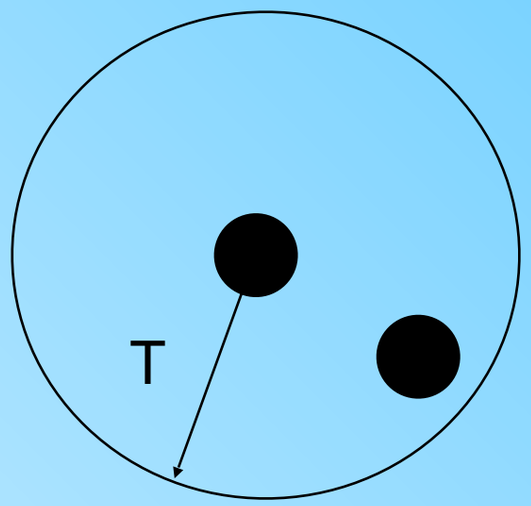
Param	Value	Desc
B	2	Branching Factor determines the maximum children allowed for a non-leaf node
T	0.15	Threshold is an upper limit to the radius of a cluster in a leaf node.
L	2	Number of Entries in a Leaf Node

Create a new cluster



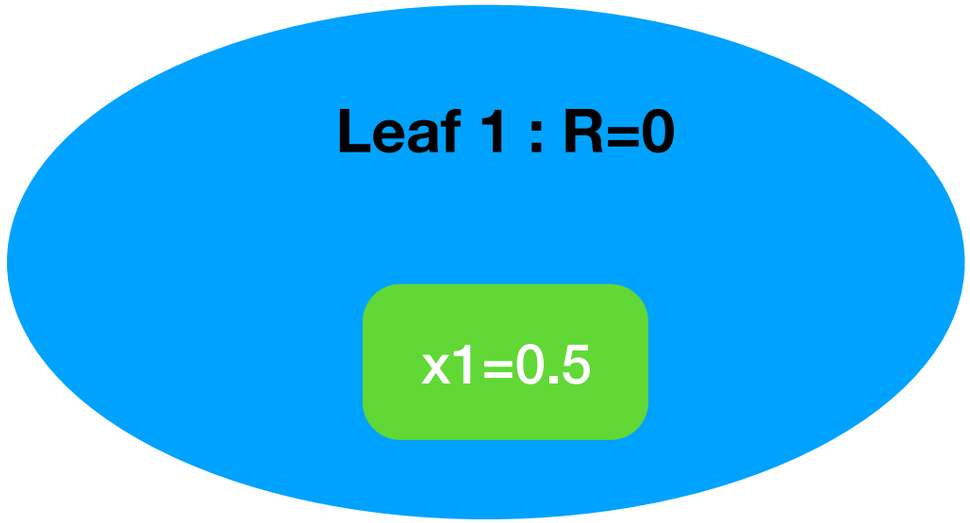
Radius: average distance from any point of the cluster to its centroid

R = sqrt(sum(xi - c)^2/ n)



If the radius of the cluster including the new record does not exceed the Threshold T, then the incoming record is assigned to that cluster

Row	x
1	0.5



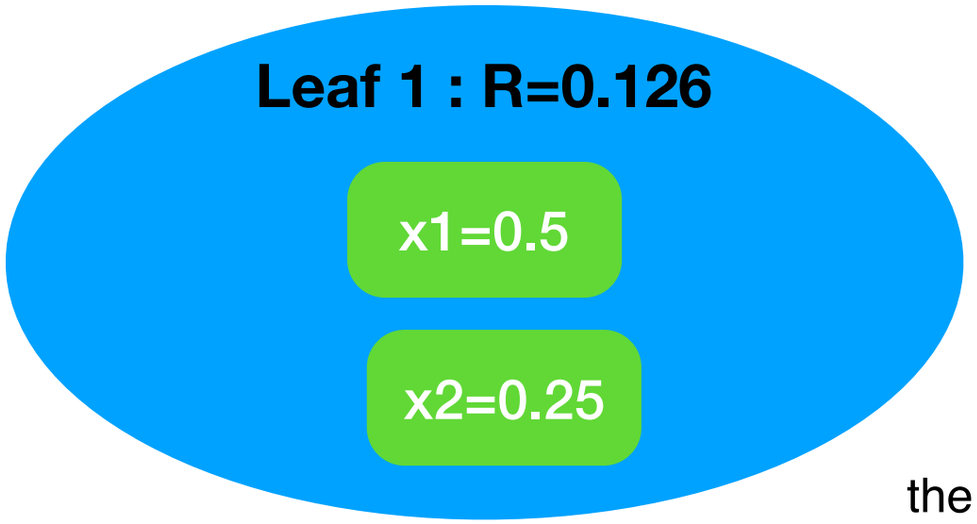
the radius is zero, and thus

less than T=0.15

Centroid c = Sum(x) / Number of data points = 0.5 /1 = 0.5

R = sqrt(sum(xi - c)^2/ n) = sqrt(sum(0.5 - 0.5)^2 / 1 = 0

Row	x
1	0.5
2	0.25



the radius is 0.125,

and thus less than T=0.15

Centroid c = Sum(x) / Number of data points = (0.5 + 0.25) / 2 = 0.375

R = sqrt(sum(xi - c)^2/ n)

R = sqrt(((0.5 - 0.375)^2 + (0.25 - 0.375)^2)/2) = 0.125

Cluster Featuring

BIRCH clustering achieves its high efficiency by clever use of a small set of summary statistics to represent a larger set of data points.

A CF tree represents a compressed form of the data, preserving any clustering structure in the data.

A CF is a set of three summary statistics that represent a set of data points in a single cluster.

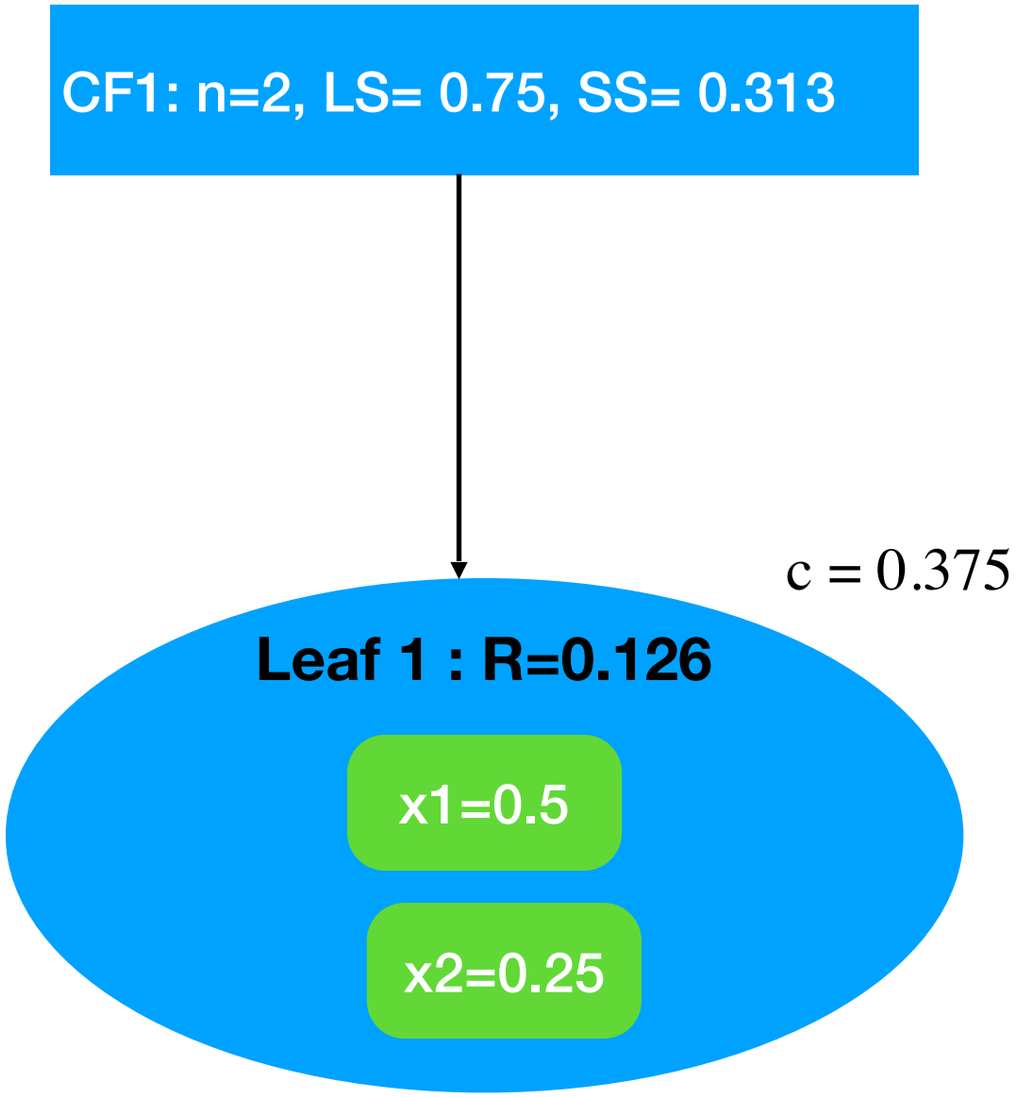
Clustering Feature (CF): CF = (N, LS, SS)

N: Number of data points

LS: linear sum of N points Sum(x)

SS: square sum of N points Sum(xA2)

Row	x
1	0.5
2	0.25

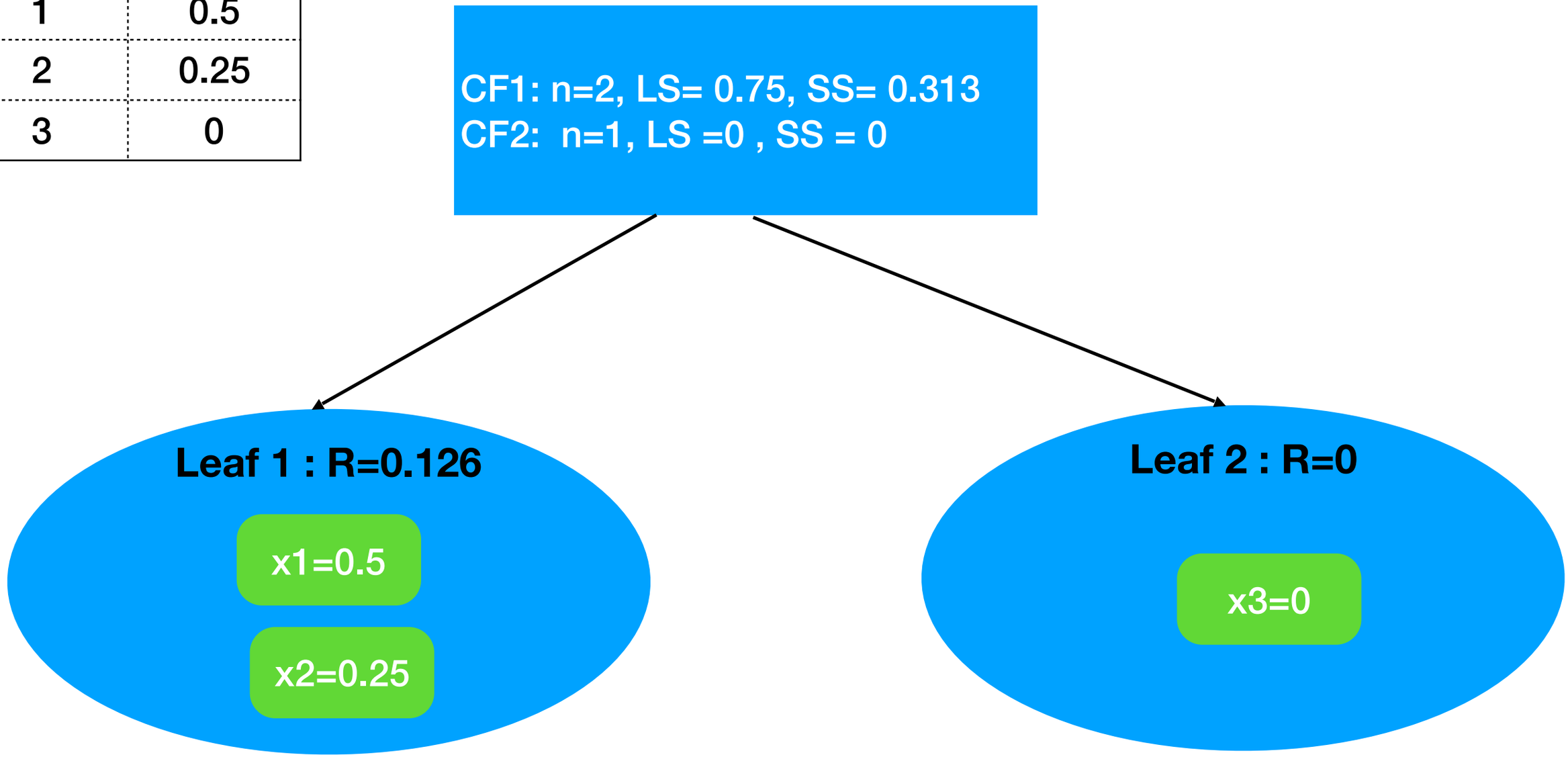


LS: linear sum of N points — Sum(x) = 0.5 + 0.25 = 0.75

SS: square sum of N points. — Sum(x^2) = 0.5^2 + 0.25^2 = 0.313

Centroid c = Sum(x) / Number of data points — c = (0.5+ 0.25)/ 2 = 0.375

Row	x
1	0.5
2	0.25
3	0



For x3 = 0, R= 0.205 >T = 0.15

so x3 is assigned to a new Leaf

Cluster Center

Center = LS/n

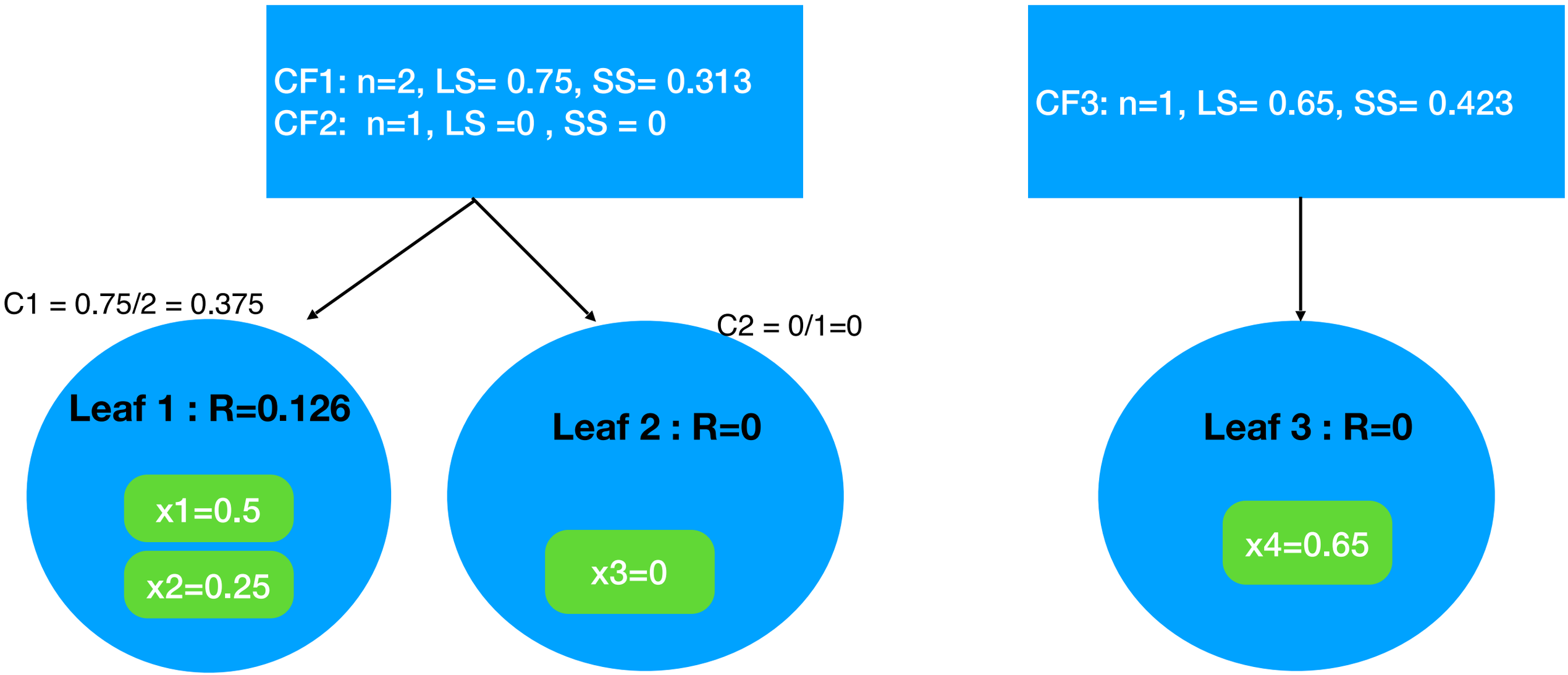
LS: linear sum of N points Sum(x)

Row	x
1	0.5
2	0.25
3	0
4	0.65

For x4 = 0.65, compares x4 to the locations of CF1 and CF2.

X4 is thus closer to CF1.

R = 0.166 > T =0.15. we would like to initialize a new leaf. However L = 2 means that we cannot have three leafs in a leaf node.We must therefore split the root node.



Cluster Feature addition

eg

Cluster 1

(2,5)

(3,2)

(4,3)

$CF1= \langle 3, (2+3+4 , 5+2+3), (22+32+42 , 52+22+32)\rangle = \langle 3, (9,10), (29 ,38)\rangle$

$CF2= \langle 3, (35,36), (417 ,440)\rangle$

$CF3=CF1+CF2= \langle 3+3, (9+35, 10+36), (29+417 , 38+440)\rangle = \langle 6, (44,46), (446 ,478)\rangle$

CF12: n=3, LS= 0.75, SS= 0.313
CF3: n=1, LS =0.65 , SS = 0.423

CF1: n=2, LS= 0.75, SS= 0.313
CF2: n=1, LS =0 , SS = 0

CF3: n=1, LS= 0.65, SS= 0.423

cf1 = 0.75/2 = 0.375

cf2 = 0/1=0

Leaf 1 : R=0.126

x1=0.5

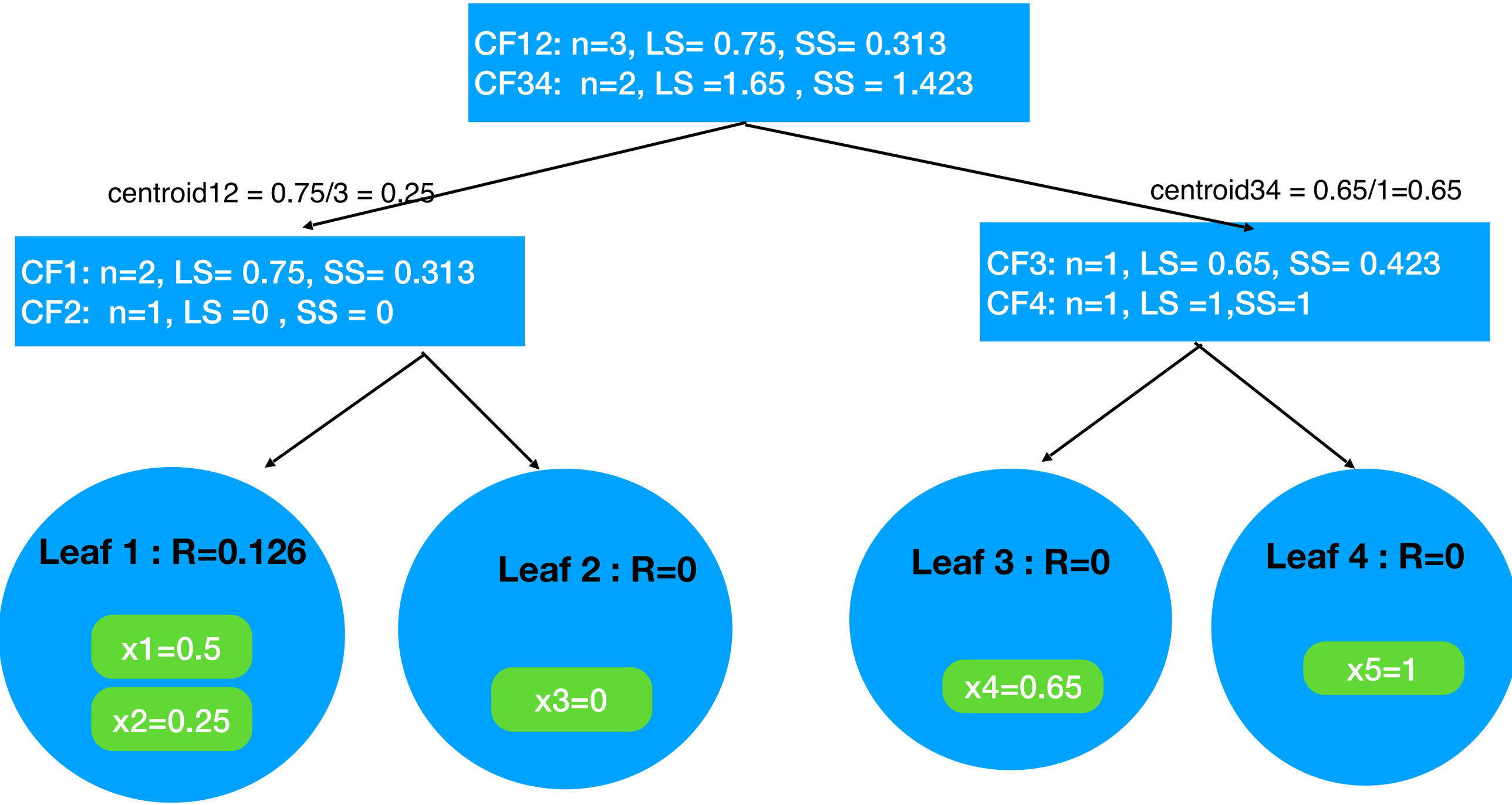
x2=0.25

Leaf 2 : R=0

x3=0

Leaf 3 : R=0

x4=0.65



For x5 = 1, BIRCH compares x5 to the locations of CF12 and CF34.

X5 is thus closer to CF34.

R = 0.175 > T =0.15. initialize a new leaf.

CF12: n=2, LS= 0.75, SS= 0.313
CF345: n=3, LS =3.05 , SS = 3.393

Center = 0.75/2=0.375

Center = 3.05/3=1.016

CF1: n=2, LS= 0.75, SS= 0.313
CF2: n=1, LS =0 , SS = 0

CF34: n=2, LS= 1.65, SS= 1.423
CF5: n=1, LS =1.4,SS=1.96

Leaf 1 : R=0.126

x1=0.5

x2=0.25

Leaf 2 : R=0

x3=0

CF3: n=1, LS= 0.65, SS= 0.423
CF4: n=1, LS =1,SS=1

CF5: n=1, LS= 1.4, SS= 1.96

Leaf 3 : R=0

x4=0.65

Leaf 4 : R=0

x5=1

Leaf 5 : R=0

x6=1.4

Center = LS/n

CF12: n=2, LS= 0.75, SS= 0.313
CF345: n=3, LS =3.05 , SS = 3.393

x7=1.1

Center = 0.75/2=0.375

Center = 3.05/3=1.016

CF1: n=2, LS= 0.75, SS= 0.313
CF2: n=1, LS =0 , SS = 0

CF34: n=2, LS= 1.65, SS= 1.423
CF5: n=1, LS =1.4,SS=1.96

Center = 1.65/2=0.825

Leaf 1 : R=0.126

x1=0.5

x2=0.25

Leaf 2 : R=0

x3=0

CF3: n=1, LS= 0.65, SS= 0.423
CF4: n=1, LS =1,SS=1

Center = 1.4/1=1.4

CF5: n=1, LS= 1.4, SS= 1.96

Leaf 3 : R=0

x4=0.65

Leaf 4 : R=0

x5=1

Leaf 5 : R=0

x6=1.4

Center = LS/n

CF12: n=2, LS= 0.75, SS= 0.313
CF345: n=4, LS =4.15 , SS = 4.212

CF1: n=2, LS= 0.75, SS= 0.313
CF2: n=1, LS =0 , SS = 0

CF34: n=3, LS= 2.75, SS= 2.252
CF5: n=1, LS =1.4,SS=1.96

Leaf 1 : R=0.126

x1=0.5

x2=0.25

Leaf 2 : R=0

x3=0

CF3: n=1, LS= 0.65, SS= 0.423
CF4: n=2, LS =2.1,SS=2.21

CF5: n=1, LS= 1.4, SS= 1.96

Leaf 3 : R=0

x4=0.65

Leaf 4 : R=0

x5=1

x7=1.1

Leaf 5 : R=0

x6=1.4

x8=m

CF12: n=2, LS= 0.75, SS= 0.313
CF345: n=4, LS =4.15 , SS = 4.212

c=?

c=?

CF1: n=2, LS= 0.75, SS= 0.313
CF2: n=1, LS =0 , SS = 0

CF34: n=3, LS= 2.75, SS= 2.252
CF5: n=1, LS =1.4,SS=1.96

Leaf 1 : R=0.126

c=?

Leaf 2 : R=0

c=?

CF3: n=1, LS= 0.65, SS= 0.423
CF4: n=2, LS =2.1,SS=2.21

CF5: n=1, LS= 1.4, SS= 1.96

Leaf 3 : R=0

c=?

Leaf 4 : R=0

c=?

Leaf 5 : R=0

c=?

In the second step, the algorithm scans all the leaf entries in the initial CF tree to rebuild a smaller CF tree, while removing outliers and grouping crowded subclusters into larger ones. This step is marked optional

c=?

c=?

Leaf 3 : R=0

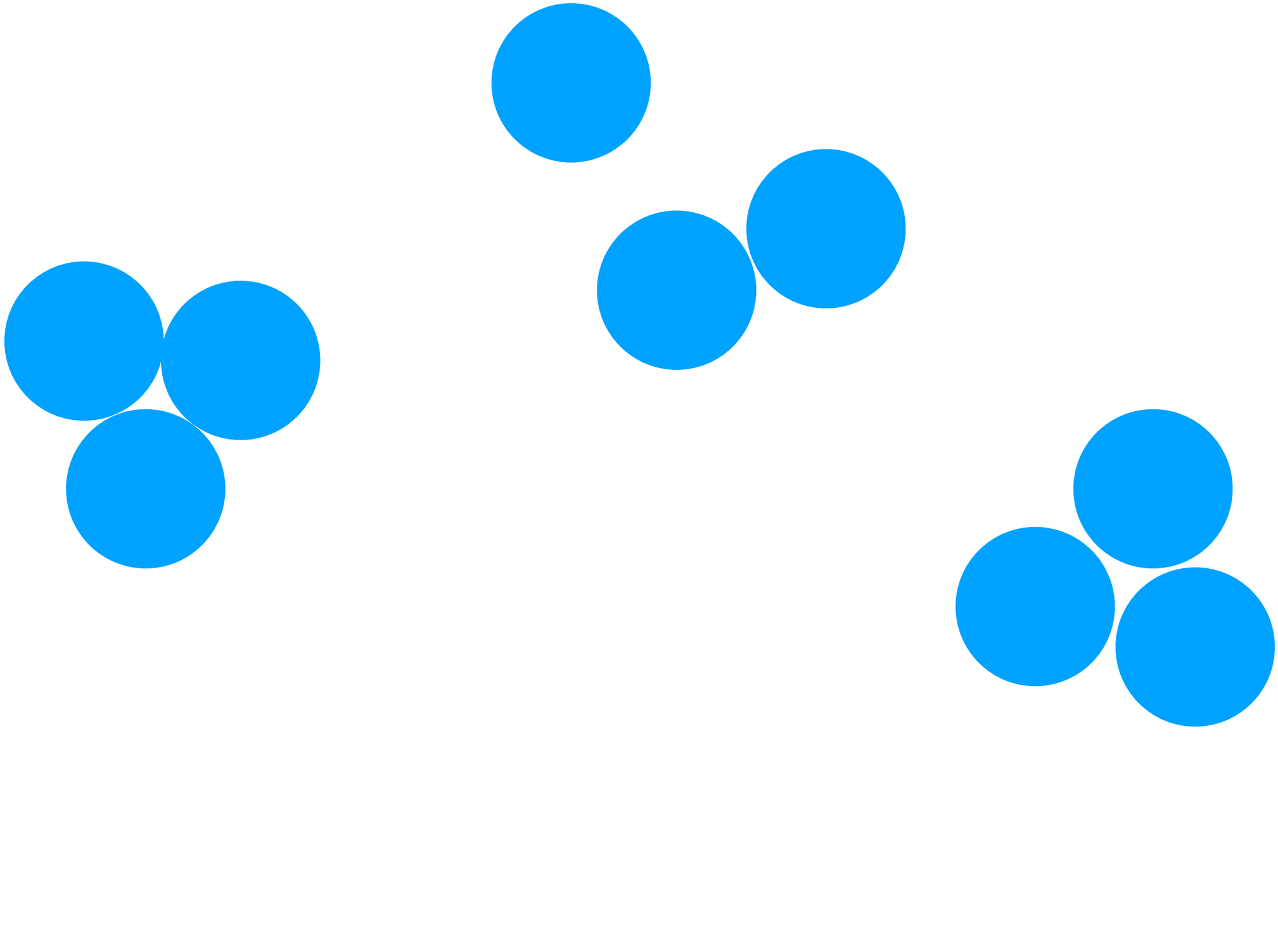
c=?

Leaf 4 : R=0

c=?

Leaf 5 : R=0

c=?



c=?

c=?

c=?

c=?

c=?

Phase 2

Once the CF tree is built, any clustering algorithm, such as a typical partitioning algorithm, can be used with the CF tree in Phase 2.

Due to the skewed input order and/or the splitting effect by the page size, the clustering structure from the initial CF-tree might not be accurate enough to reflect the real underlying clustering structure. To address this issue, the second key step ("global clustering") tries to cluster all the subclusters in the leaf nodes. This is done by converting a subcluster with n ′ data points n ′ times at the centroid and then running either an agglomerative hierarchical clustering algorithm or a modified clustering algorithm.

Phase 2 often uses agglomerative hierarchical clustering

The five CFs - CF1, CF2, CF3, CF4, CF5 are the objects that the agglomerative clustering shall be carried out on, not the original data.

The cluster centers are as follows

CF1 = 0.375, CF2 =0, CF3=0.65, CF4 = 1.05, CF5 =1.4

The two closest clusters are CF1 and CF3. Combining these CFs, we obtain a new cluster center

CF13 = (2 * 0.375 + 1 * 0.65)/3 = 0.47

The remaining cluster centers are

CF13 = 0.47, CF2 =0, CF4 = 1.05, CF5 =1.4

Appendix

Multi-class classification with growing number of classes

An example for such a task could be classifying if an image shows a dog or a cat. Furthermore, the model should be able to recognize that a goose doesn't fit into one of these two categories, thus create a new class.

Rather than new "classes" it should rather be not of the class matching the training data. You would want to actually consider there One-Class Classification (or Single-Class Classification) where the one-class would be defined as both dogs and cats in your training example. This is also called Outlier Detection.

https://stats.stackexchange.com/questions/278892/multi-class-classification-with-growing-number-of-classes-question

a benefit of BIRCH clustering is that the analyst is not required to select the best choice of k, the number of clusters, as is the case with some other clustering methods. Rather, the number of clusters in a BIRCH clustering solution is an outcome of the treebuilding process

Diameter: square root of average mean squared distance between all pairs of points in the cluster

D = sqrt(sum(sum(xi - xj) ^2) / n)

Experimental Results

KMEANS clustering

DS	Time	D	# Scan	DS	Time	D	# Scan
1	**43.9**	**2.09**	**289**	1o	33.8	1.97	197
2	13.2	4.43	51	2o	12.7	4.20	29
3	32.9	3.66	187	3o	36.0	4.35	241

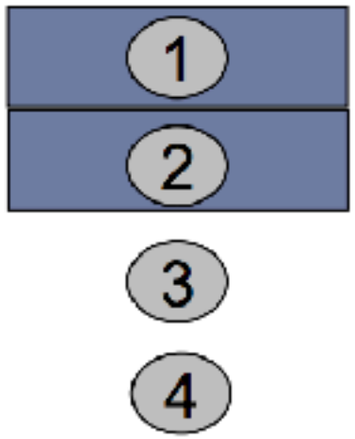
BIRCH clustering

DS	Time	D	# Scan	DS	Time	D	# Scan
1	**11.5**	**1.87**	**2**	1o	13.6	1.87	2
2	10.7	1.99	2	2o	12.1	1.99	2
3	11.4	3.95	2	3o	12.2	3.99	2

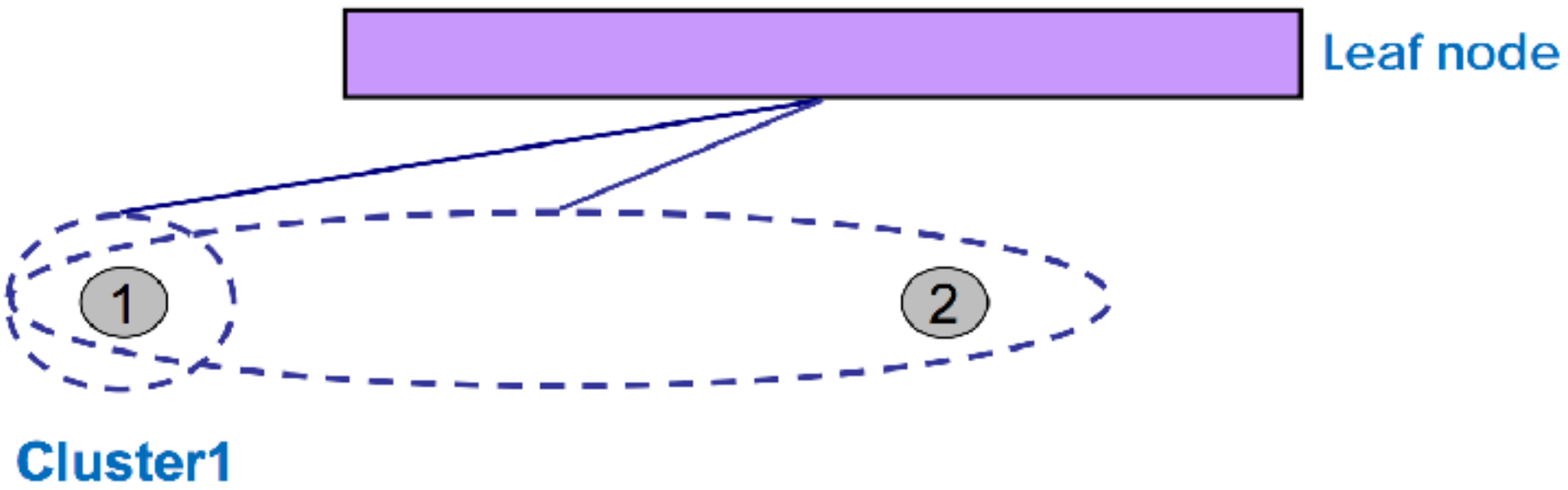
definition

- An unsupervised data mining algorithm used to perform hierarchical clustering over particularly large data-sets.

Data Objects

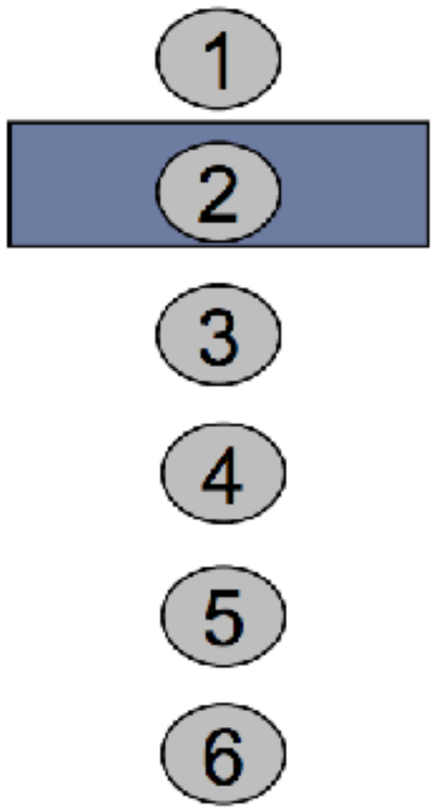


Clustering Process (build a tree)

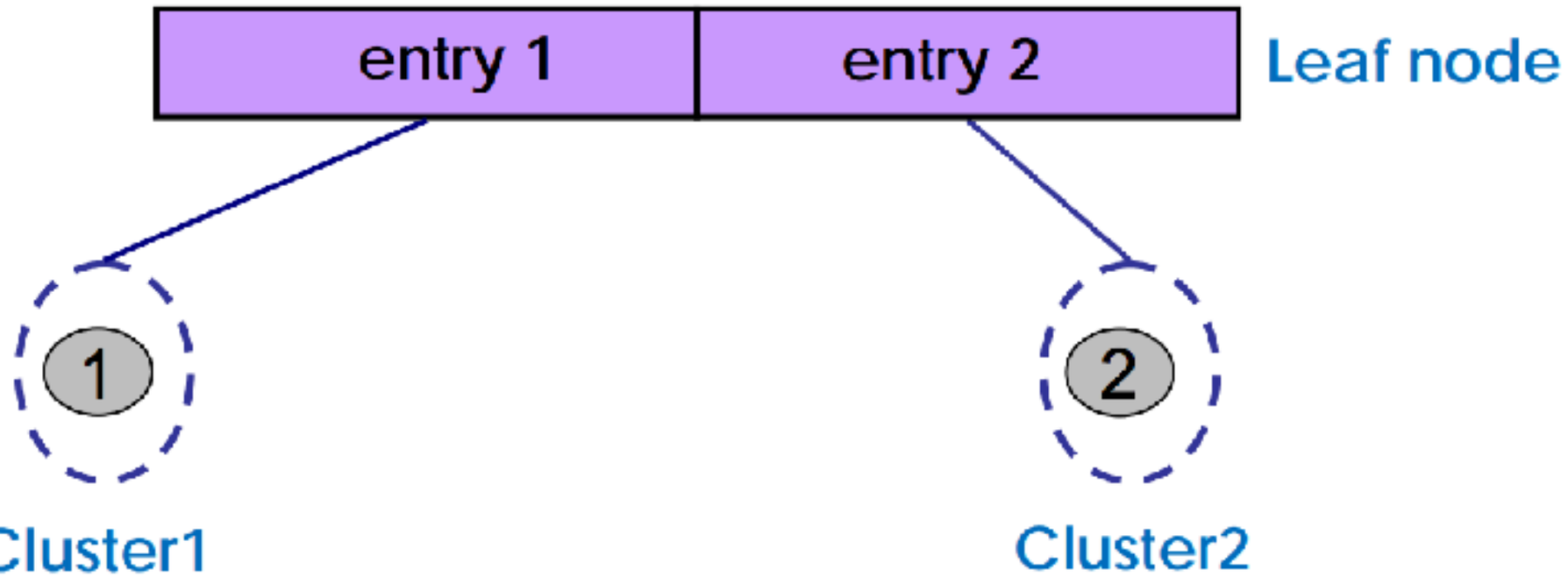


If cluster 1 becomes too large (not compact) by adding object 2, then split the cluster

Data Objects

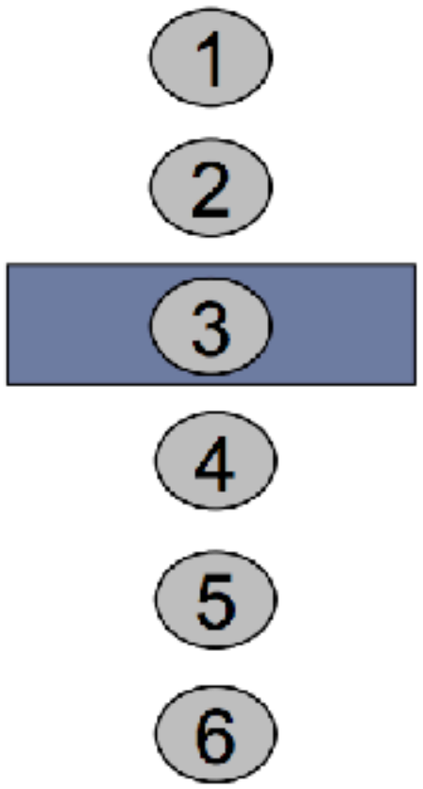


Clustering Process (build a tree)

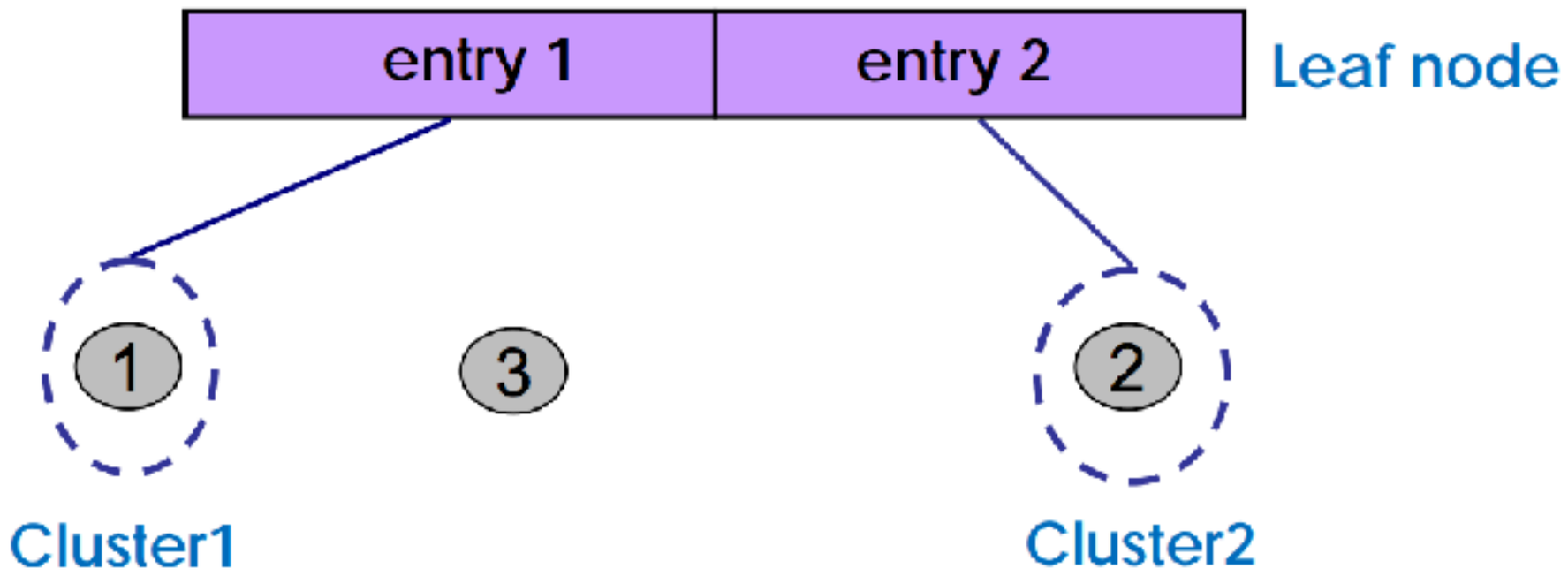


Leaf node with two entries

Data Objects



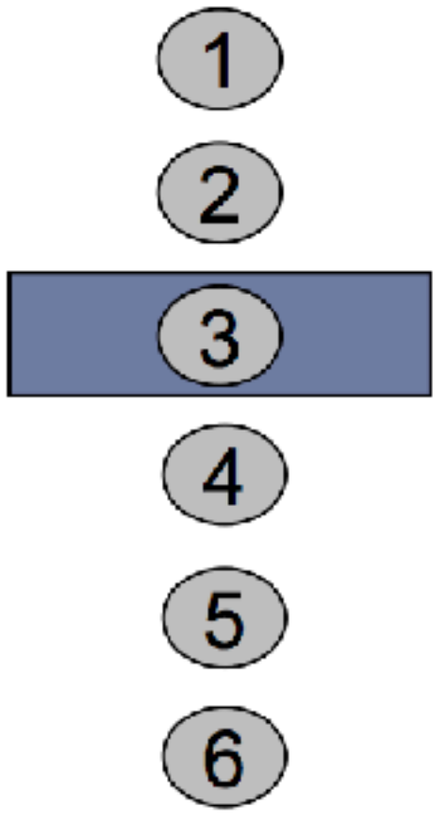
Clustering Process (build a tree)



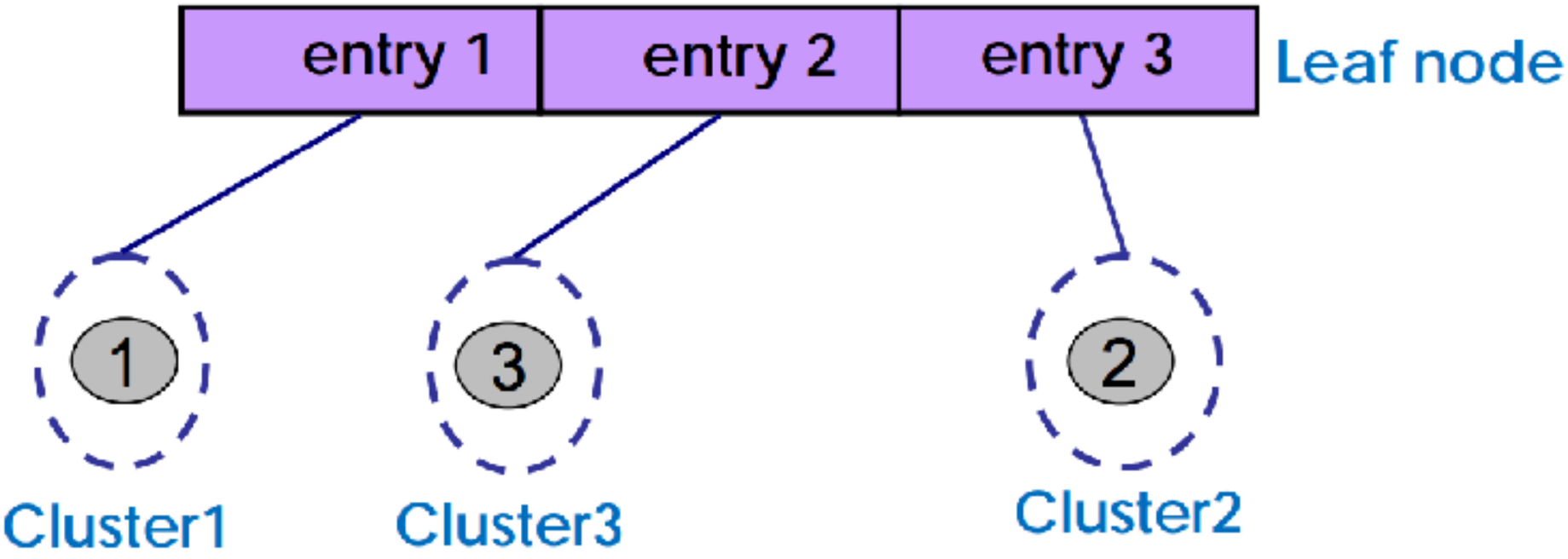
entry1 is the closest to object 3

If cluster 1 becomes too large by adding object 3, then split the cluster

Data Objects

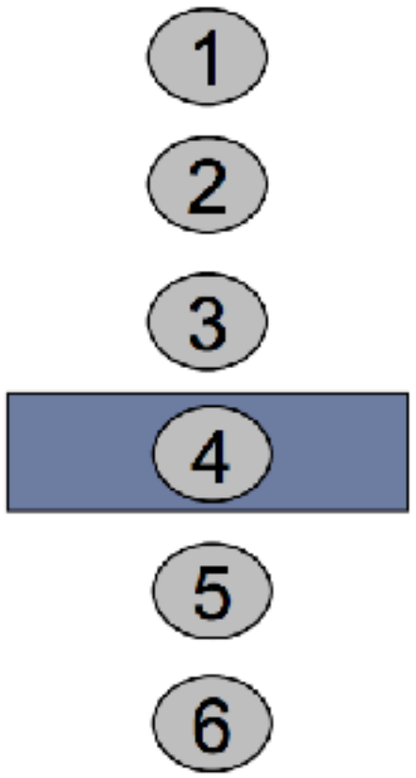


Clustering Process (build a tree)

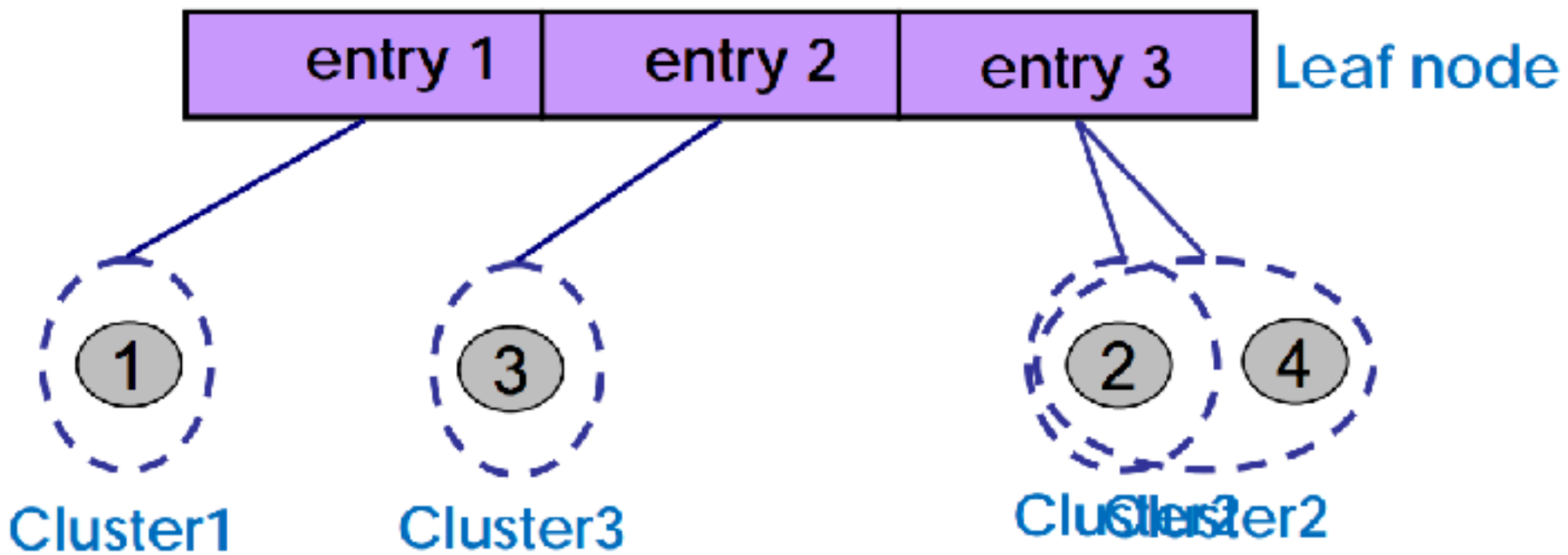


Leaf node with three entries

Data Objects

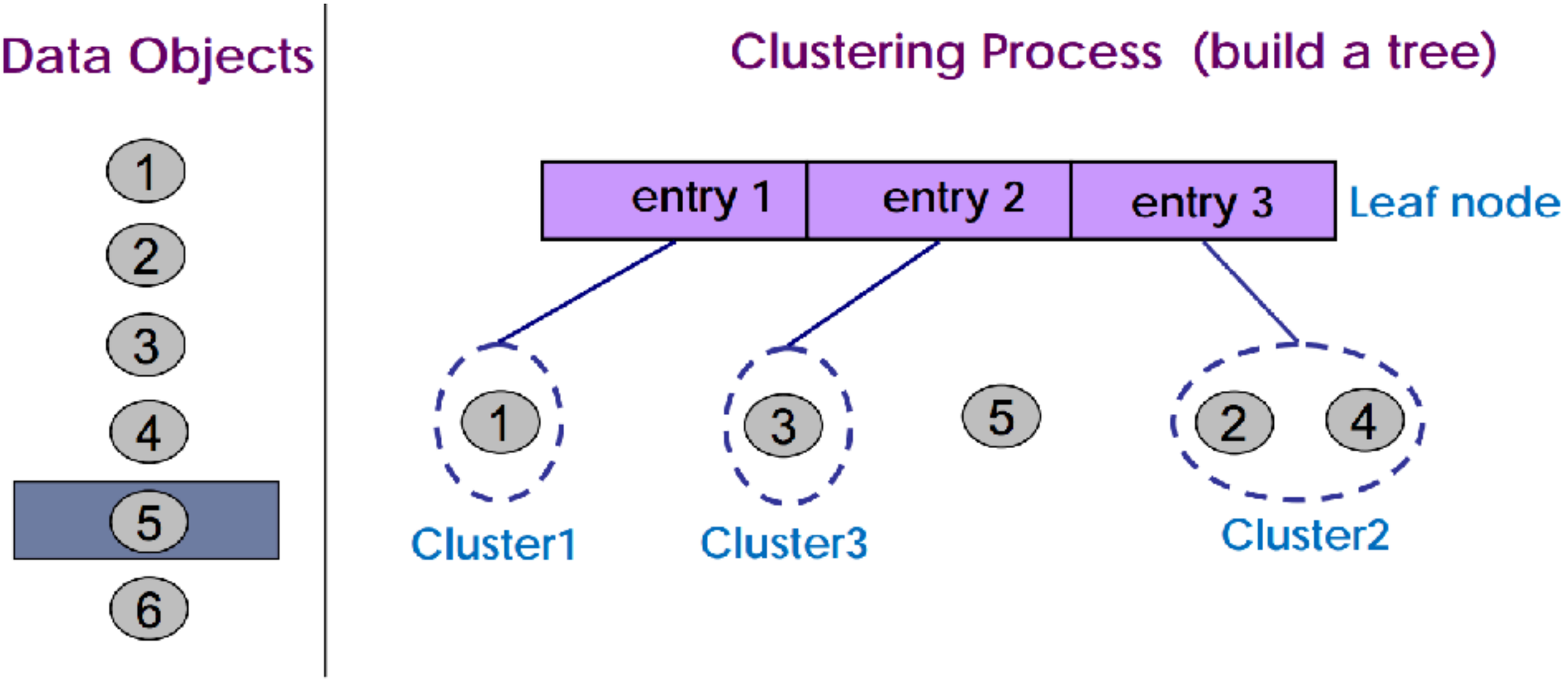


Clustering Process (build a tree)



entry3 is the closest to object 4

Cluster 2 remains compact when adding object 4
then add object 4 to cluster 2



Cluster 2 is the closest to object 5

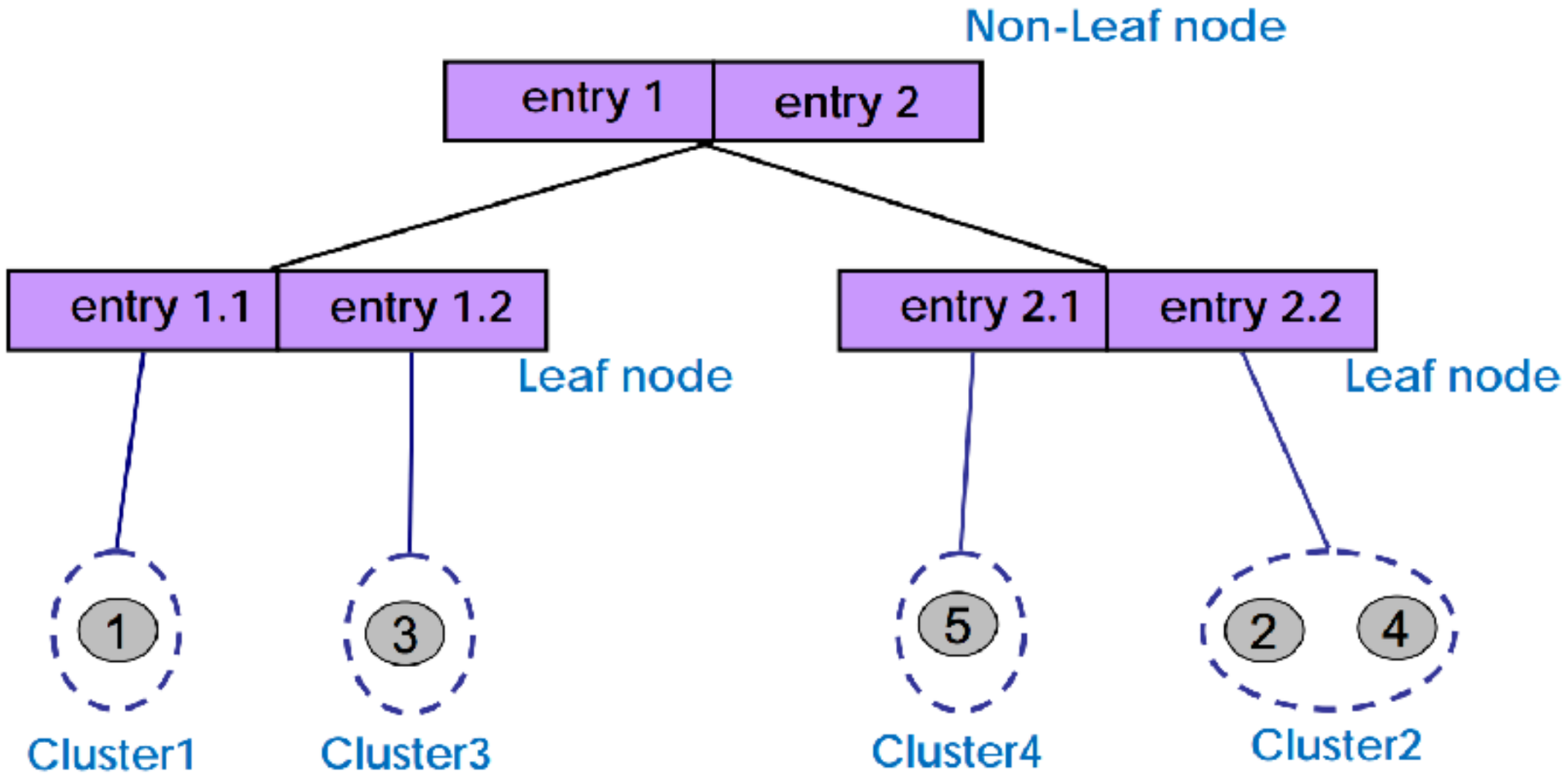
Cluster 2 becomes too large by adding object 5 then split cluster 2?

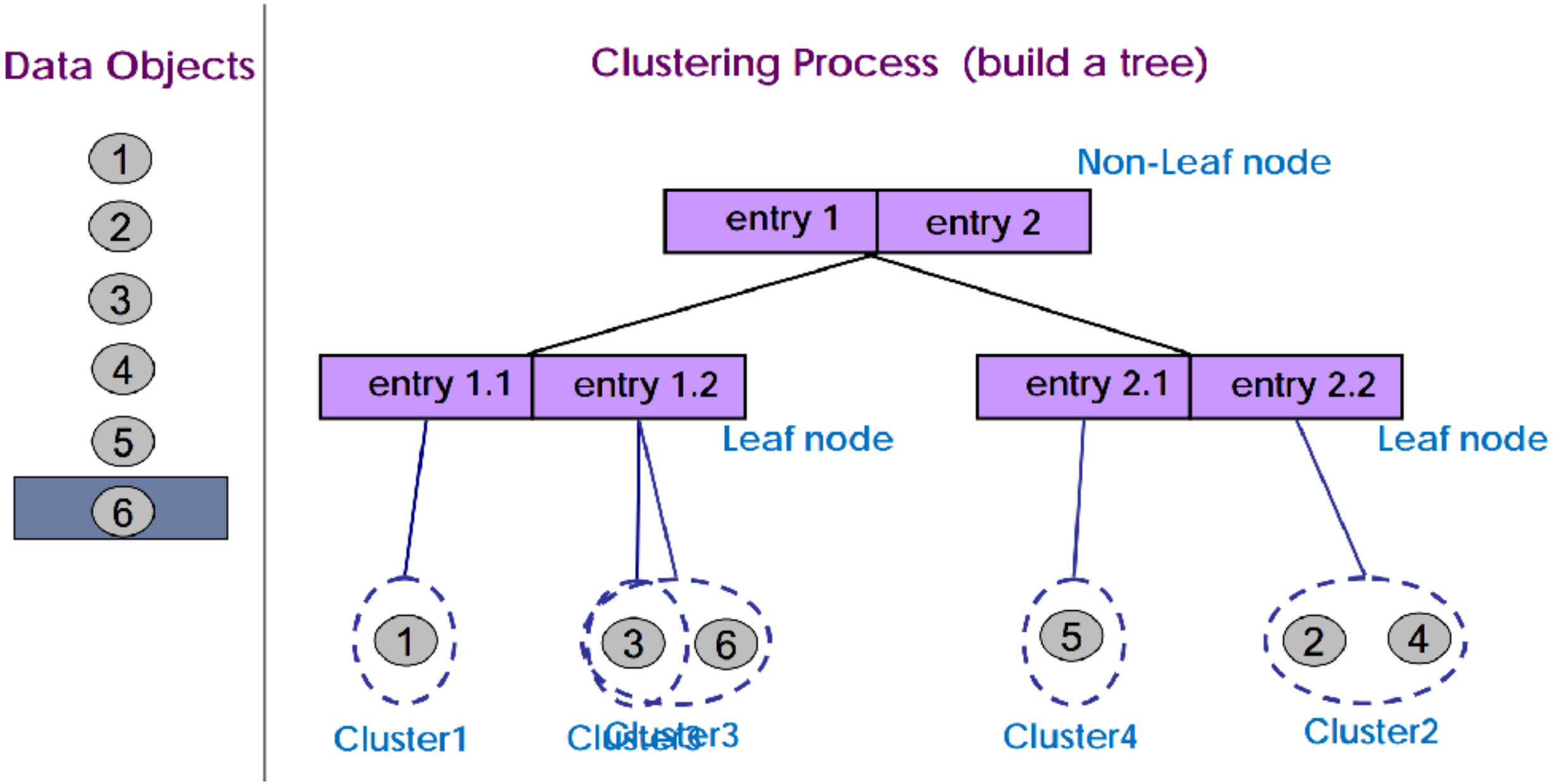
BUT there is a limit to the number of entries a node can have Thus, split the node

Data Objects

1
2
3
4
5
6

Clustering Process (build a tree)





entry1.2 is the closest to object 6

Cluster 3 remains compact when adding object 6

then add object 6 to cluster 3

Exam Questions

- ***What is the main limitation of BIRCH?***
  - *Since each node in a CF tree can hold only a limited number of entries due to the size, a CF tree node doesn't always correspond to what a user may consider a nature cluster. Moreover, if the clusters are not spherical in shape, it doesn't perform well because it uses the notion of radius or diameter to control the boundary of a cluster.*